

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À  
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE  
DE LA MAÎTRISE EN GÉNIE ÉLECTRIQUE

PAR  
MAXIME BOULAY

ÉTUDE COMPARATIVE DES TECHNIQUES D'ANNULATION DES  
INTERFÉRENCES DANS UN SYSTÈME MIMO-OFDM

DÉCEMBRE 2009

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

## Résumé

Le monde des communications est présentement à un carrefour très important de son évolution. La prochaine génération (quatrième) des systèmes sans fil emploiera une modulation basée sur le multiplexage par répartition orthogonale de la fréquence (OFDM - Orthogonal Frequency Division Multiplexing) qui est actuellement utilisée dans les routeurs sans fil type WiFi. De plus, afin de s'approcher des performances des systèmes avec fil, plusieurs antennes à l'émission et plusieurs antennes à la réception (MIMO – Multiple Input Multiple Output) seront utilisées.

Dans le cadre de notre recherche, nous nous sommes concentrés sur le récepteur, plus précisément sur le détecteur. Nous avons fait une étude comparative de différentes techniques d'égalisation MIMO-OFDM. Ce faisant, avons aussi adapté des techniques d'égalisation qui n'avaient pas encore été utilisées en MIMO-OFDM dans la littérature, afin d'élargir leur champ d'utilisation. L'étude que nous avons réalisée s'est traduite par une recherche sur les techniques en question, puis la construction d'une plate-forme de simulation, et enfin, de la simulation des techniques. Les résultats valident en bonne partie des informations retrouvées dans la littérature.

Nous ne pouvons recommander une seule technique, car cela dépend toujours des besoins et circonstances, mais la technique appelée SQRD a définitivement attiré notre attention en combinant des performances respectables et une relative simplicité au niveau des calculs.

## Remerciements

Avant tout, j'aimerais remercier mon directeur de recherche, M. Adel Omar Dahmane. Je le remercie tant pour l'aide à la recherche elle-même, que pour la dimension humaine de la maîtrise, puisqu'il m'a donné un support important du début à la fin, tant dans les creux de vague que dans les meilleurs moments.

Je tiens aussi à remercier certains professeurs du département qui m'ont aidé à différents moments de ma maîtrise, comme M. Kodjo Agbossou et M. Mamadou Lamine Doumbia.

Merci aussi à mes amis, tant les compagnons de maîtrise que ceux hors du contexte scolaire, qui permettent de parfois penser à autre chose.

Enfin, un grand merci à ma famille pour son support inconditionnel dans mon entreprise.

## Table des matières

Résumé.....	ii
Remerciements.....	iii
Table des matières.....	iv
Liste des tableaux.....	vii
Liste des figures .....	viii
Liste des acronymes .....	ix
Chapitre 1 - Introduction.....	1
1.1 Problématique.....	3
1.2 Objectifs .....	6
1.2.1 Objectif principal .....	6
1.2.2 Objectifs secondaires .....	6
1.3 Méthodologie.....	6
1.4 Organisation du mémoire .....	8
Chapitre 2 - Système MIMO-OFDM.....	10
2.1 Système OFDM ( <i>orthogonal frequency division multiplexing</i> ).....	10
2.2 Fonctionnement d'un système OFDM .....	12

2.3	Fonctionnement d'un système MIMO.....	15
2.4	Système MIMO-OFDM .....	17
2.4.1	Transmetteur MIMO-OFDM.....	18
2.4.2	Récepteur MIMO-OFDM .....	19
Chapitre 3 - Techniques de réception MIMO-OFDM .....		21
3.1	<i>Zero-forcing</i> (ZF) .....	22
3.2	<i>Minimum Mean-Squared Error</i> (MMSE) .....	24
3.3	V-BLAST ( <i>Vertical Bell Laboratories Layered Space-Time</i> ).....	25
3.4	<i>Least mean-square</i> (LMS).....	28
3.5	<i>Recursive least-squares</i> (RLS) .....	30
3.6	<i>Zero-Forcing Sorted QR decomposition</i> (ZF-SQRD).....	32
3.7	<i>Minimum Mean-Squared Error Sorted QR decomposition</i> (MMSE-SQRD) .....	37
3.8	<i>Simplified sphere decoding</i> (SSD).....	39
Chapitre 4 - Simulations et résultats .....		47
4.1	Plate-forme de simulation.....	47
4.2	Simulations des techniques adaptatives.....	50
4.3	Simulation des techniques non adaptatives .....	55
4.4	Complexité algorithmique .....	60

4.5 Analyse finale et recommandations d'égaliseurs .....	61
Chapitre 5 - Conclusion .....	64
5.1 Retour sur le contenu .....	64
5.2 Recommandations .....	65
Références .....	67

## Liste des tableaux

Tableau 1 : Complexité des algorithmes.....	61
---	----



## Liste des figures

Figure 1 : Exemple de quatre fréquences orthogonales .....	11
Figure 2 : Exemple d'un système OFDM typique .....	13
Figure 3 : Constellation 16-QAM .....	14
Figure 4 : Exemple d'un récepteur typique OFDM .....	15
Figure 5 : Système multi-antennes.....	16
Figure 6 : Transmetteur MIMO-OFDM.....	18
Figure 7 : Récepteur MIMO-OFDM.....	19
Figure 8 : Algorithme du SD.....	41
Figure 9 : Exemple de recherche de point du SD montrant les points inspectés .....	44
Figure 10 : Algorithme du SSD .....	45
Figure 11 : BER du LMS .....	51
Figure 12 : BER du RLS .....	52
Figure 13 : Performances du LMS selon le nombre d'antennes .....	53
Figure 14 : Performances du RLS selon le nombre d'antennes .....	53
Figure 15 : Performances du ZF et du MMSE.....	56
Figure 16 : Performances des algorithmes V-BLAST .....	57
Figure 17 : Performances des algorithmes SQRD .....	58
Figure 18 : Comparaison d'égaliseurs en configuration 2X4 .....	59

## Liste des acronymes

3G :	3 <sup>e</sup> génération des systèmes de communication sans-fil
4G :	4 <sup>e</sup> génération des systèmes de communication sans-fil
BER:	<i>Bit error rate</i>
CDMA :	<i>Code division multiple access</i>
CP:	<i>Cyclic prefix</i>
DSP:	<i>Digital signal processor</i>
FFT:	<i>Fast Fourier transform</i>
IBI :	<i>Interblock interference</i>
ICI :	<i>Interchannel interference</i>
IFFT:	<i>Inverse fast Fourier Transform</i>
ISI :	<i>Intersymbol interference</i>
LMS:	<i>Least mean square</i>
MIMO :	<i>Multi-input multi-output</i>
ML:	<i>Maximum likelihood</i>
MMSE:	<i>Minimum mean-squared error</i>
MSE:	<i>Mean-squared error</i>

OFDM :	<i>Orthogonal frequency division multiplexing</i>
QAM:	<i>Quadrature amplitude modulation</i>
RLS:	<i>Recursive least square</i>
SIC:	<i>Successive interference cancellation</i>
SD:	<i>Sphere decoding</i>
SNR:	<i>Signal-to-noise-ratio</i>
SSD:	<i>Simplified sphere decoding</i>
V-BLAST :	<i>Vertical Bell Laboratories layered Space-Time</i>
ZF:	<i>Zero-forcing</i>

## Chapitre 1 - Introduction

Le monde des communications est présentement à un carrefour très important de son évolution. C'est particulièrement vrai en ce qui concerne les communications sans fil, dont l'importance ne cesse de s'accroître très rapidement à cause des nombreuses avancées dans le domaine, et de leur accessibilité au grand public. De plus, grâce à la miniaturisation des technologies, leur performance a augmenté, et continuera d'augmenter à un rythme effréné. Les secteurs de la communication sans fil et de la communication mobile constituent en ce sens des défis importants. Au fil du temps, plusieurs générations se sont succédé afin d'améliorer le débit et la capacité tout en maintenant une qualité de service appréciable. Le monde sans fil est ainsi passé par la modulation analogique avant d'adopter définitivement la modulation numérique.

L'évolution dans ce domaine se fait très rapidement : une nouvelle génération voit à peine le jour que des recherches sont faites sur la prochaine. La raison en est fort simple : l'augmentation du nombre d'utilisateurs de services sans-fil est presque exponentielle, et chacun de ces utilisateurs demande toujours plus de bande passante. En effet, à l'origine, on se contentait de transmettre le strict nécessaire pour qu'il y ait communication, c'est-à-dire la voix. Toutefois maintenant, on transmet maintenant des données réseau, de la voix, l'internet, des fichiers audio, vidéo... Nous voyons donc que l'évolution de ce domaine n'est pas prête de s'arrêter, ni même de ralentir.

Durant toute cette évolution, la base reste la même qu'elle a toujours été dans les systèmes de télécommunications : nous désirons émettre et recevoir, tout en limitant les interférences et les effets perturbateurs du canal de transmission. Les problèmes restent donc aussi en bonne partie les mêmes : le bruit gaussien, les interférences (intersymbole, les délais...) Toutefois, selon les méthodes utilisées pour la communication [RAP96], [HAY01], [YAC02] la façon de résoudre les différents problèmes varie grandement, car les techniques ne sont souvent applicables que pour un mode d'envoi précis, ou bien elles s'appliquent différemment selon le mode.

Dans le cadre de ce projet, nous nous concentrerons sur les égaliseurs (souvent aussi appelés détecteurs). C'est la partie du récepteur qui doit éliminer le bruit du signal, afin de le reconstituer le plus fidèlement possible. Pour ce faire, on utilise les canaux estimés préalablement par l'estimateur de canal [COL02], [OUA02]. Les égaliseurs doivent aussi faire appel à différentes méthodes de filtrage [HAY96], [SAY03].

Au cours des dernières années, des systèmes utilisant la modulation OFDM (orthogonal frequency division multiplexing [BAH04], [SUN01]) ont été développés et perfectionnés; ils sont présentement utilisés dans les normes 802.11a, g et 802.16. Les systèmes OFDM sont particulièrement appréciés parce qu'ils résistent bien aux déformations causées par le canal. La détection de signal en est donc simplifiée [KAN04], [LIU03].

À l'OFDM, nous désirons jumeler les avantages des systèmes à plusieurs antennes. Nous appelons ces systèmes MIMO (multi-input multi-output) [GOL03], [KUR04], [ZHE03], et ils sont étudiés de plus en plus. Il s'agit d'utiliser plusieurs antennes à l'émission et à la réception, afin de maximiser le nombre de canaux différents, ce qui permettrait de diminuer le taux d'erreurs ou d'augmenter le débit du transfert de données,

selon la configuration choisie. Toutefois, la majorité des travaux concernant le MIMO sont d'une nature théorique, et le MIMO a comme principal défaut de requérir une grande puissance de calcul au niveau du détecteur [ALD01], [BOR04], [WIN06], [ZAN05].

On en arrive donc aux systèmes de télécommunications MIMO-OFDM [BOL02], [LAN04], [RAO04], [UTH04], [YAN05], qui devraient composer, en parallèle avec le CDMA [HON02], [SMI02], [SUR06], [TAC02], la 4<sup>e</sup> génération de systèmes de communication sans-fil (4G). La raison de la dualité, c'est que les systèmes CDMA (*code division multiple access*; accès multiple par répartition en code) de la 3<sup>e</sup> génération (3G) ne pourront pas être remplacés rapidement, car des millions d'utilisateurs l'utilisent présentement. Les deux technologies devront donc coexister pendant plusieurs années. Des études sont même faites sur des technologies hybrides [ELB05], [HAR97], [YAN06] combinant les deux.

### 1.1 Problématique

Les systèmes MIMO-OFDM sont au coeur de nombreuses recherches en télécommunications. Comme tous les autres systèmes de communication, ils sont affectés par des interférences, que nous désirons éliminer. Évidemment, en pratique, aucun système ne peut complètement les éliminer; l'objectif est donc de les minimiser. Ces interférences sont de plusieurs types. Tout d'abord, il y a le simple bruit blanc, que nous retrouvons dans tous les systèmes et qui est causé par la chaleur, les composantes électriques environnantes, etc. Ensuite, comme de nombreux symboles sont transmis les uns après les autres, et que les différents chemins (effet de chemins multiples; *multipath*) qu'ils empruntent causent des délais différents, les échos de symboles déjà envoyés viennent interférer sur les symboles présents. C'est ce que nous appelons l'interférence intersymbole (*intersymbol interference*;

ISI). Son principal effet est un évanouissement (*fading*) du signal. Si les délais entre la réception des différentes versions du signal sont très longs, nous pouvons aussi retrouver de l'interférence interbloc (*interblock interference*; IBI). Enfin, comme nous traitons dans ce mémoire d'un système utilisant plusieurs porteuses, l'interférence intercanal (*interchannel interference*; ICI) peut apparaître si le système n'est pas parfaitement configuré.

Plusieurs solutions sont possibles pour éliminer, ou plutôt minimiser l'effet de ces interférences. La première est un choix judicieux du type de système de télécommunications : nous avons mentionné en introduction que nous allons travailler sur les égaliseurs pour un système MIMO-OFDM. La modulation OFDM est reconnue pour sa grande résistance à l'ISI, grâce à l'utilisation de fréquences orthogonales pour envoyer les différents symboles. De plus, l'utilisation du MIMO vient accroître l'efficacité en offrant plusieurs autres chemins possibles pour le signal, augmentant la fiabilité des transmissions. Le MIMO-OFDM coexistera donc avec le MIMO-CDMA dans la quatrième génération, car avec autant d'utilisateurs, le CDMA ne peut disparaître rapidement...

Toutefois, peu importe l'efficacité reconnue de la méthode utilisée, il est important d'avoir un bon égaliseur pour profiter pleinement des avantages d'une technique. Comme les systèmes MIMO-OFDM sont à l'étude, et que peu de prototypes existent présentement, nous en sommes toujours à vouloir déterminer quel algorithme d'égalisation utiliser. Une étude comparative s'impose donc, afin de faire ressortir les forces et les limites de chacun.

Les techniques d'égalisation MIMO-OFDM sont nombreuses [LEU03], [LI05], [MED06], [SON05], [TAN05] et se présentent sous la forme de différents types [WAN05] (différentes classifications sont possibles) :

- Linéaires non adaptatives

- Non linéaires non adaptatives
- Linéaires adaptatives
- Non linéaires adaptatives

Dans les techniques linéaires non adaptatives, nous retrouvons la technique la plus simple, soit celle du *Zero-Forcing* (ZF). Elle consiste à éliminer l'interférence due au canal seulement (note : nous assumons les canaux comme étant connus, ou parfaitement estimés). Toutefois, ce faisant, elle ne considère pas le bruit, et tend à en amplifier l'effet. Cette technique ne semble donc pas très efficace, mais comme l'algorithme est de loin le plus simple, ce n'est pas une surprise. Pour remédier à son problème, la méthode *Minimum Mean-Squared Error* (MMSE) a été développée, basée sur le ZF. Elle a pour but de minimiser l'erreur quadratique moyenne, afin de reconstituer le signal. Nous augmentons alors d'un cran la complexité.

Pour ce qui est des techniques non linéaires, nous notons le V-BLAST, qui consiste à trouver le récepteur dont le signal a le plus de chances d'avoir la bonne information. Ensuite, l'effet de cette antenne est annulé pour calculer, de la même manière, la prochaine antenne avec le meilleur signal, et ainsi de suite. Des techniques de ZF et de MMSE peuvent être utilisées pour déterminer quelles antennes considérer, résultant en ZF-VBLAST et MMSE-BLAST. Une autre méthode non linéaire est celle du *Maximum Likelihood* (ML), qui offre d'excellentes performances, au prix d'une très grande complexité au niveau des calculs.

Dans les techniques adaptatives, nous pouvons retrouver l'algorithme du gradient stochastique (*Least Mean Square*; LMS) et celui des moindres carrés récurrents (*Recursive*



*least square*; RLS), ainsi que plusieurs autres. Nous devons toutefois calibrer ces algorithmes avec une séquence d'apprentissage, et bien que leurs performances soient très bonnes, leur complexité n'est pas négligeable, surtout dans des techniques telles que le RLS.

## **1.2 Objectifs**

### *1.2.1 Objectif principal*

L'objectif principal de ce mémoire est d'effectuer une étude comparative de plusieurs méthodes d'égalisation pour des systèmes de télécommunications de type MIMO-OFDM. Le but est d'en arriver à la recommandation de l'une des méthodes analysées.

### *1.2.2 Objectifs secondaires*

- Répertorier et classer les différents types d'algorithmes.
- Établir une base de comparaison qui permet de classer les algorithmes selon leurs performances et leur complexité.
- Créer une plate-forme qui permet la simulation des algorithmes. Une seule plate-forme doit être créée, afin de comparer tous les algorithmes sur un pied d'égalité.
- Adapter des techniques d'égalisation qui n'ont pas encore été utilisées pour le MIMO-OFDM.
- Faire ressortir les principales caractéristiques des différentes méthodes.

## **1.3 Méthodologie**

La première étape consiste en une recherche bibliographique approfondie, afin de répertorier les différents égaliseurs traités dans la littérature. Cette recherche se poursuivra tout au long du projet, car il est très possible qu'une méthode conduise à une autre que nous

n'avions pas encore trouvée, ou bien que nous devons trouver des informations additionnelles sur une méthode lorsque viendra le temps de sa simulation.

Avant d'aller directement à la simulation, des étapes restent à faire. Tout d'abord, nous devons, à l'aide des résultats retrouvés dans la littérature, identifier des critères permettant la comparaison. Il y a deux catégories de critères à considérer : performance et complexité. Si les critères sont bien choisis, nous pourrons aussi ensuite faire un rapport performance/complexité qui permettra de déterminer quelle méthode est réellement la plus efficace en terme d'utilisation de ressources.

Ensuite, nous approchons de la simulation. Toutefois, comme les techniques d'égalisation à simuler sont nombreuses, et que l'un des points importants de ce projet est de faire la comparaison dans des conditions similaires, il faut d'abord développer une plate-forme de base, sous MATLAB, où nous pourrons tester les différentes méthodes. Le but de la plate-forme, c'est de ne plus avoir à s'occuper de la génération de signaux OFDM, de la création du canal, etc. Seule la méthode d'égalisation sera différente, ce qui nous permettra d'en tirer des conclusions significatives. La plate-forme doit bien sûr être en mesure de faire ressortir les résultats en fonction des critères de comparaison déterminés précédemment. Une base intéressante est présentée dans [WAN04], et servira à la conception de la plate-forme.

Une fois la plate-forme développée, nous pourrons coder, puis simuler chacune des techniques d'égalisation, afin d'en sortir les courbes et résultats pertinents pour notre étude comparative. Différentes conditions (canaux, signaux envoyés, etc.) pourront être testées, mais pour une même comparaison, elles seront toujours les mêmes. Par exemple, nous pourrions faire une étude comparative des techniques en 64-QAM, puis une autre en 256-

QAM; il est possible que les résultats de comparaison ne soient pas les mêmes sous certaines conditions, ce qui serait intéressant à observer.

Une fois les résultats sortis en fonction des critères déterminés, il faudra en faire l'analyse, afin de faire la comparaison des techniques, et en fonction des conditions choisies. Il serait aussi intéressant d'être en mesure de justifier, à l'aide des équations, pourquoi une méthode est bel et bien la meilleure sous certaines conditions, mais pas dans d'autres. Cette analyse servira donc aussi à mettre en évidence les caractéristiques principales de chacune des techniques étudiées.

Enfin, nous allons recommander une méthode, selon ce qui aura ressorti de l'analyse. En fait, il est possible de recommander plus d'une méthode si les résultats varient beaucoup en fonction des conditions.

#### **1.4 Organisation du mémoire**

Dans le chapitre 2, nous expliquons plus précisément ce que sont le MIMO et l'OFDM, pour en arriver au MIMO-OFDM. Ensuite, nous regardons les composantes du système de télécommunications avec lequel nous allons travailler.

Au chapitre 3, nous allons décrire en détail les techniques qui seront comparées dans le cadre de l'analyse comparative. Nous allons tenter d'en faire ressortir les caractéristiques telles que décrites dans la littérature.

Le chapitre 4 débute en expliquant la plate-forme de simulation MATLAB, où se déroulent toutes les simulations. De nombreuses simulations des techniques seront faites et présentées, et bien sûr, nous allons faire l'analyse des résultats.

Enfin, le chapitre 5 est celui de la conclusion. Nous ferons un retour sur les objectifs ainsi qu'un rapide retour sur quelques points importants de ce mémoire, avant de terminer sur des recommandations pour de futurs travaux.

## Chapitre 2 - Système MIMO-OFDM

Dans ce chapitre, nous décrivons ce qu'est un système MIMO-OFDM, afin de voir les bases du système auquel cette recherche s'intéresse. Nous aborderons une étape à la fois, soit d'abord les systèmes OFDM, puis le MIMO, et enfin la combinaison de ces deux éléments.

### 2.1 Système OFDM (*orthogonal frequency division multiplexing*)

Les bases de l'OFDM ont été jetées dans les années 1960. Toutefois, il aura fallu attendre les années 1990 et la présence croissante des DSP (processeur de signal numérique; *digital signal processor*) avant d'y voir un réel intérêt. Jusque-là, les techniques de télécommunications sans fil se résumaient généralement à un signal envoyé en utilisant une porteuse, puis capté par une antenne pour en récupérer l'information. Bien sûr, des développements technologiques ont amélioré cette forme de télécommunications, mais la base du principe, soit l'utilisation d'une porteuse, n'a pas changé en plusieurs décennies.

De tels systèmes sont toutefois sujets à l'interférence intersymbole (ISI). Celle-ci se produit surtout lorsque la durée d'un symbole est courte par rapport au délai de transmission. Les différents échos empruntant différents trajets se mélangent et compliquent la détection lors de la réception.

La nouveauté apportée par le système OFDM est l'utilisation de plusieurs sous-porteuses, de fréquences orthogonales, réparties sur une certaine bande de fréquences. L'orthogonalité des fréquences est très importante, car c'est elle qui assure l'absence d'interférences entre les sous-porteuses (ICI). La performance de ce système dépend donc beaucoup de la synchronisation des fréquences entre le transmetteur et le récepteur, afin d'assurer l'orthogonalité, mais ce n'est normalement pas problématique. Il suffit que l'espacement entre les fréquences soit de  $1/T_s$ , où  $T_s$  est la durée d'un symbole, pour que les fréquences soient orthogonales, telles que les quatre illustrées dans la Figure 1.

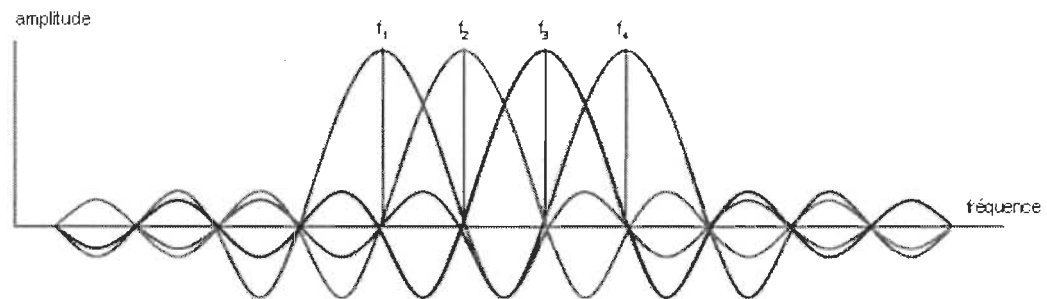


Figure 1 : Exemple de quatre fréquences orthogonales

L'orthogonalité des fréquences assure donc qu'une porteuse n'empiètera pas sur l'autre, car elle devrait être nulle à la fréquence des autres porteuses.

Dans un système OFDM, l'ISI est beaucoup plus faible que dans un système à une seule porteuse. En effet, la transmission de plusieurs informations en parallèle permet d'envoyer ces informations plus lentement pour un même débit. Cela nous permet d'avoir une durée de symboles rallongée par rapport au délai de transmission, et c'est ce qui diminue l'ISI. Un intervalle de garde est généralement ajouté entre les symboles, toujours afin de minimiser l'ISI le plus possible.

Un autre avantage inhérent aux systèmes OFDM est le fait que nous pouvons considérer les canaux sélectifs en fréquence non pas comme une fonction de la fréquence, mais plutôt comme une constante pour une sous-porteuse donnée. La raison est que la largeur de bande d'une sous-porteuse est très petite, ce qui fait qu'effectivement, la variation rattachée aux canaux sélectifs en fréquence est minimale, et pratiquement négligeable. Cette hypothèse simplifiera beaucoup les calculs, et le système dans son ensemble, tout en restant très près de la réalité.

La méthode la plus utilisée pour transformer des valeurs numériques (le signal à transmettre) en de multiples sous-porteuses est l'application du principe de la FFT (*fast Fourier transform*; transformée de Fourier rapide) et sa réciproque, l'IFFT. Durant les dernières décennies, l'algorithme de la FFT a été optimisé à un point tel que c'est maintenant une opération de base intégrée à la majorité des composantes de traitement de signal, telle que les DSP (*digital signal processor*, processeur de signal numérique).

## **2.2 Fonctionnement d'un système OFDM**

Maintenant que nous avons énuméré plusieurs caractéristiques des systèmes OFDM, voyons la mécanique qui se cache derrière. Nous allons nous baser sur la représentation d'un système OFDM simple à 2 sous-porteuses illustrée à la figure 2.

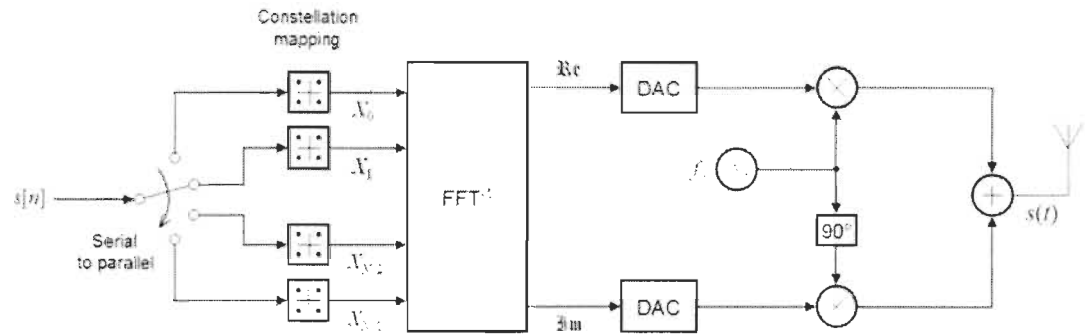


Figure 2 : Exemple d'un système OFDM typique

Sur la figure 2, nous avons toutes les étapes de base d'un système OFDM typique (tiré de Wikipedia), pour un système à une seule antenne de transmission.  $s(n)$  contient la série de données à transmettre. Les deux premières étapes sont interchangeables dans leur ordre. L'une d'elle consiste à effectuer une opération de démultiplexage, afin de répartir la série de données en parallèle. L'autre étape consiste à appliquer la constellation choisie au signal d'entrée.

Dans ce mémoire, nous utiliserons les constellations QAM (*quadrature amplitude modulation*; modulation d'amplitude en quadrature) rectangulaires, telles que le 16-QAM illustré à la figure 3.

Bien qu'elles n'optimisent pas l'espacement entre les différents symboles pour une énergie précise, les constellations QAM sont très utilisées pour leur simplicité. Pour cette raison, ce sont celles que nous retrouvons le plus souvent dans les ouvrages et articles scientifiques, et c'est ce qui a motivé le choix de les utiliser.



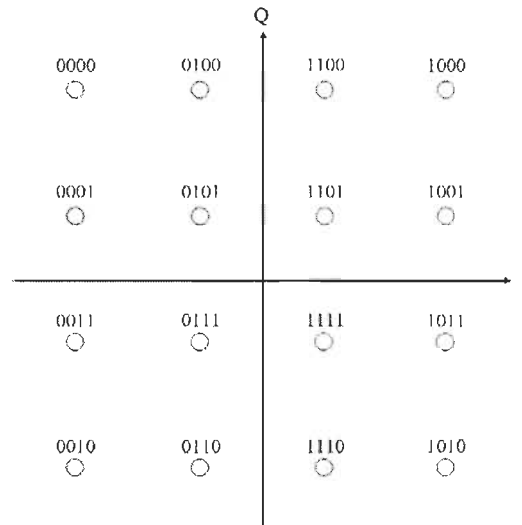


Figure 3 : Constellation 16-QAM

Une fois les données encodées selon la constellation, nous appliquons une IFFT (notée  $\text{FFT}^{-1}$  sur le schéma). Nous traitons donc les données d'entrée comme si elles étaient dans le domaine fréquentiel, et que nous voulions les ramener au domaine temporel. Le nombre de points de l'IFFT correspond au nombre de sous-porteuses (deux dans le schéma de la Figure 2).

Comme nous pouvons nous y attendre, le récepteur OFDM refait les opérations inverses pour retrouver le signal d'origine, comme l'illustre le schéma de la figure 4 :

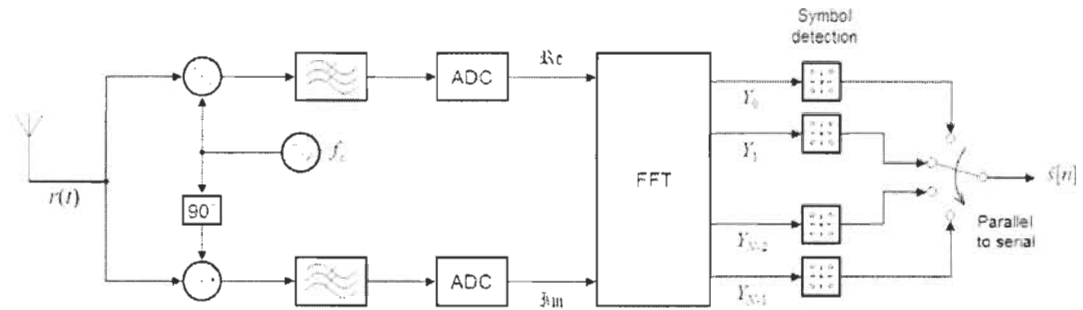


Figure 4 : Exemple d'un récepteur typique OFDM

La première partie consiste en une démodulation de base : multiplication et filtre passe-bas. Nous appliquons ensuite une FFT afin de retourner dans le domaine fréquentiel. Pour terminer, la détection de symbole est appliquée selon la constellation utilisée, puis les données repassent de parallèle à série. Comme pour la transmission, ces deux dernières étapes sont aussi interchangeables.

Ce que nous venons de voir n'est bien sûr qu'un système OFDM de base. Toutefois, nous recroiserons les principes énumérés plus loin, lorsque nous parlerons de la plate-forme utilisée. Passons maintenant au MIMO.

### 2.3 Fonctionnement d'un système MIMO

Dans cette partie, nous allons voir le fonctionnement d'un système MIMO, afin de pouvoir par la suite combiner cette théorie avec celle sur l'OFDM pour décrire le système dans son ensemble.

Comme nous l'avons vu au chapitre 1, le MIMO est un système utilisant plusieurs antennes pour la transmission comme pour la réception. Le schéma de la figure 5 représente un tel système.

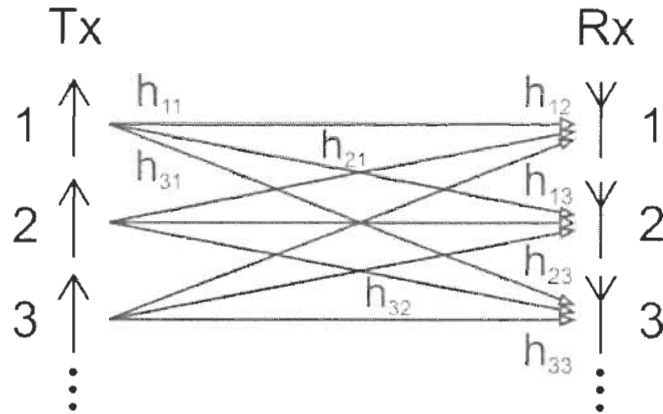


Figure 5 : Système multi-antennes

Un tel système suppose que les antennes sont suffisamment espacées, c'est-à-dire d'un minimum d'une demi longueur d'onde. En effet, si ce n'est pas le cas, du point de vue de la transmission, ça serait comme si toutes les antennes étaient au même point. Il serait donc inutile d'avoir plusieurs antennes. En effet, le but d'avoir plusieurs antennes est de créer une diversité au niveau du canal. Le canal n'est donc plus unique. Nous le représentons alors par la matrice canal  $\mathbf{H}$ . Pour un système comptant  $N_t$  antennes pour la transmission et  $N_r$  antennes pour la réception, la matrice sera de taille  $N_r \times N_t$ . Les éléments  $h_{nrm}$  de cette matrice représentent alors le canal tel que perçu par l'antenne de réception  $n_r$  pour un signal provenant de l'antenne de transmission  $n_t$ . Ceci étant dit, nous pouvons exprimer un tel système MIMO par cette simple équation :

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v} \quad (2.1)$$

où  $\mathbf{x}$  est le vecteur  $N_t \times 1$  du signal transmis,  $\mathbf{y}$  le vecteur  $N_r \times 1$  du signal reçu, et  $\mathbf{v}$  le bruit gaussien additif. Ce dernier est ajouté à l'entrée du récepteur pour représenter les inévitables interférences amassées en chemin.

Nous observons au passage qu'il s'agit en fait d'une généralisation des systèmes de télécommunications SISO, dont l'équation est, à un moment précis :

$$y = hx + v \quad (2.2)$$

C'est-à-dire qu'un système SISO n'est qu'un cas particulier du système MIMO, où le nombre d'antennes à la transmission et à la réception ( $N_t$  et  $N_r$ ) est égal à un. Cette observation est intéressante, car nous pourrons ainsi facilement comparer le MIMO avec le SISO simplement en changeant le nombre d'antennes.

## 2.4 Système MIMO-OFDM

Maintenant que nous avons exploré les concepts de systèmes MIMO et OFDM, nous pouvons plonger dans le cœur de notre sujet, c'est-à-dire ceux qui combinent ces deux notions : les systèmes MIMO-OFDM. Il s'agit en fait simplement d'un système OFDM appliqué sur plusieurs antennes qui transmettent des informations parallèles. Le grand défi auquel nous faisons face est donc de retrouver les données envoyées, à partir du mélange d'informations reçues. Nous élaborerons sur les techniques de réception permettant d'y arriver au prochain chapitre.

Nous commencerons d'abord par définir les principales composantes du système MIMO-OFDM. Comme tout système de télécommunications, celui-ci est constitué d'un transmetteur, d'un canal, et d'un récepteur, qui sont eux-mêmes composés de quelques éléments.

### 2.4.1 Transmetteur MIMO-OFDM

Avant d'aller plus loin, mentionnons qu'il existe deux types principaux de transmission MIMO. Ces deux choix mettent en évidence un problème courant en télécommunications : choisir entre un débit plus élevé ou bien en une meilleure qualité de transmission. En effet, dans un système MIMO, nous pourrions choisir d'envoyer, avec chaque antennes, la même information, afin d'augmenter encore la diversité des canaux. Toutefois, cette technique est peu utilisée, et il n'en sera pas question dans ce travail. Nous considérons donc que les antennes de transmission envoient des informations différentes, que nous voudrions reconstruire avec le récepteur. Nous pouvons voir sur la figure 6 les différentes parties du transmetteur MIMO-OFDM.

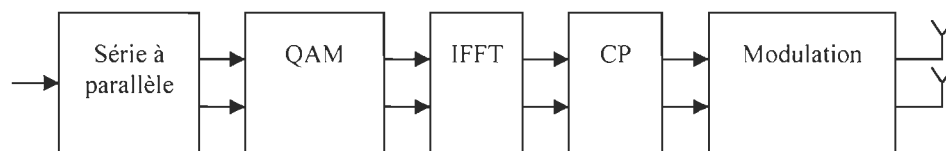


Figure 6 : Transmetteur MIMO-OFDM

La forme générale est très semblable au transmetteur OFDM illustré à la figure 2. Nous la représentons ici différemment pour mettre en évidence les éléments MIMO, et pour se rapprocher de la plate-forme que nous verrons en détails au chapitre 4. Le schéma illustre 2 antennes de transmission, mais ce nombre pourra varier.

La chaîne de données d'entrée est, à l'origine en série. La première étape sera donc de faire passer ces données en parallèle, afin de les répartir sur autant de séries qu'il y aura d'antennes de transmission.

Les données ainsi mises en parallèle seront ensuite encodées selon une constellation QAM. À ce point, nous avons, à l'entrée de ce bloc, des valeurs de  $[0, \dots, N_{QAM} - 1]$ , où  $N_{QAM}$  est le nombre de symboles possibles dans la constellation que nous avons choisie (64 pour du 64-QAM, par exemple). Celles-ci sont ensuite associées à leur valeur complexe correspondante de la constellation.

Nous appliquons ensuite l'IFFT sur ces valeurs, donnant pour sortie des valeurs complexes. Un préfixe cyclique (*cyclic prefix*; CP) est apposé au début de chaque groupe de données OFDM, contenant une copie des dernières données. C'est à ce point que le signal est réparti sur les antennes de transmission et pour être envoyé.

#### 2.4.2 Récepteur MIMO-OFDM

Voici maintenant à quoi ressemble un récepteur MIMO-OFDM (figure 7) :

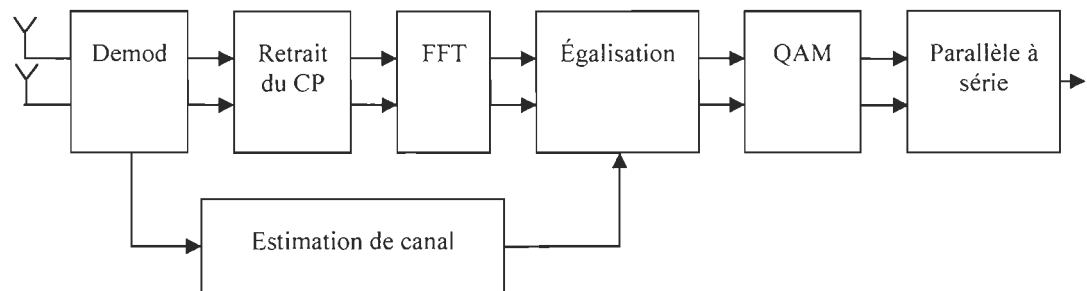


Figure 7 : Récepteur MIMO-OFDM

Le signal reçu est d'abord démodulé, afin de retrouver des données numériques. Ensuite, nous retirons le CP qui a été apposé, car il n'a aucune valeur en tant qu'information. Puis nous appliquons la FFT pour retourner dans le domaine fréquentiel. C'est ensuite que l'égalisation a lieu. Par contre, pour la plupart des techniques, il est

nécessaire de connaître le canal pour pouvoir procéder à la détection. C'est pourquoi nous retrouvons, dans ces systèmes, un estimateur de canal. Cette estimation peut se faire avec des pilotes ajoutés aux signaux, par exemple. La détection donne donc une estimation des données qui ont été envoyées à l'origine. Pour terminer, celles-ci se font attribuer leur valeur réelle en fonction de la constellation qui avait été choisie au départ. C'est ainsi que se déroule, en résumé, le trajet du signal.

Nous venons donc de nous familiariser davantage avec les concepts de MIMO et d'OFDM. Cette connaissance nous a permis d'enchaîner avec une présentation du système MIMO-OFDM qui sera utilisé dans le reste de ce document.

## Chapitre 3 - Techniques de réception MIMO-OFDM

Dans ce chapitre, nous allons détailler plusieurs des techniques d'égalisation MIMO-OFDM que nous avons mentionnées lors du tour d'horizon du chapitre 1. Rappelons les différents types de techniques qui existent.

Tout d'abord, les techniques sont définies comme étant adaptatives ou non. Le LMS et le RLS sont des exemples de techniques qui sont adaptatives. La majorité des techniques ne le sont toutefois pas. Les techniques adaptatives sont caractérisées par l'absence d'estimateur de canal dans le récepteur. Effectivement, avec ces techniques, nous n'avons pas besoin d'approximer la matrice du canal pour faire l'égalisation, contrairement aux techniques non adaptatives. Toutefois, une série de données connue à la fois du transmetteur et du récepteur doit être envoyée à intervalle régulier. Selon les altérations subies par cette série lors de son trajet, une technique adaptative se recalibrera pour compenser l'effet de canal. Plutôt que de chercher à connaître le canal lui-même, ce sont les effets qu'il a sur le signal qui sont analysés et compensés. Ces techniques ont deux principaux désavantages. Dès que nous nous écartons des techniques de base (LMS, RLS), elles ont tendance à requérir d'importantes ressources de calcul, et leur complexité croît rapidement. Aussi, le fait de devoir envoyer une séquence de calibration (*training sequence*; séquence d'entraînement) à intervalles réguliers plutôt que l'information que nous voulons transmettre ralentit un peu le transfert.



Ensuite, les techniques peuvent être linéaires ou non. La majorité des situations de télécommunications sont linéaires. Notons qu'il existe des exceptions. La plus courante dans le domaine est notamment le cas des communications satellite, qui utilisent souvent des amplificateurs en saturation pour donner la puissance nécessaire au signal lors de sa transmission. Cette saturation est un élément de non-linéarité. Si nous utilisons un égaliseur linéaire classique pour retrouver l'information envoyée, les résultats que nous obtiendrons seront au mieux médiocres.

Commençons à décrire les différentes techniques d'égalisation étudiées dans cette analyse. Avant de poursuivre, rappelons que l'équation de transmission a été donnée par l'équation 2.1.

### 3.1 *Zero-forcing (ZF)*

Le ZF est de loin la technique la plus simple et la plus instinctive. Il s'agit d'une technique d'égalisation linéaire non adaptative. Elle est basée sur le raisonnement suivant : si le signal reçu est donné par la multiplication du signal par le canal, alors nous pouvons récupérer le signal d'origine en multipliant le signal reçu par la réciproque du canal. Pour ce faire, nous voulons déterminer la matrice de poids  $\mathbf{W}$  qui permet de trouver l'estimation  $\hat{\mathbf{x}}$  du signal transmis  $\mathbf{x}$ .

$$\hat{\mathbf{x}} = \mathbf{W}\mathbf{y} \quad (3.1)$$

où

$$\mathbf{W} = \mathbf{H}^\dagger = (\mathbf{H}^* \mathbf{H})^{-1} \mathbf{H}^* \quad (3.2)$$

L'opération que nous avons ici symbolisée par une croix (+) se nomme la pseudo-inverse de Moore-Penrose. L'astérisque est la conjuguée transposée de la matrice. Comme la matrice de canal est de dimensions  $N_r \times N_t$ , elle ne sera pas nécessairement carrée, d'où l'impossibilité de faire une inversion selon la définition de base. C'est pourquoi nous employons ici cette pseudo-inverse telle que définie par l'équation 3.2. Cette opération donne d'ailleurs le même résultat que l'inverse sur les matrices qui peuvent effectivement être inversées normalement. Dans notre contexte, nous pouvons considérer que la pseudo-inverse n'est qu'une généralisation de l'inversion de matrice, plus largement applicable. D'ailleurs, comme nous le verrons, cette opération est utilisée dans plusieurs autres techniques d'égalisation.

Nous nous devons d'observer que pour cette technique d'égalisation, nous ne tenons pas compte du bruit  $\mathbf{v}$ . C'est d'ailleurs en grande partie pour cette raison que l'algorithme est aussi simple. Toutefois, il n'est pas moins présent. C'est aussi le principal inconvénient de cette technique, qui, de par son ignorance du bruit, a parfois même tendance à en amplifier les effets.

Dans la littérature, cette technique est souvent utilisée comme base de comparaison due à son extrême simplicité algorithmique. Par le fait même, ses performances sont généralement moins bonnes que les autres techniques auxquelles elle est comparée. Malgré tout, cette technique fait bel et bien une égalisation de base, et sert de pilier sur lequel viennent s'appuyer d'autres techniques plus avancées.

### 3.2 Minimum Mean-Squared Error (MMSE)

La technique MMSE est directement basée sur le ZF que nous venons de voir. Tout comme cette dernière, elle est linéaire et non adaptative. Suite à ce que nous venons de voir, nous pouvons conclure qu'une façon logique d'améliorer le ZF serait de lui faire prendre en compte l'effet du bruit. C'est exactement ce que fait la technique MMSE.

Tout comme pour le ZF, nous voulons trouver un  $\mathbf{W}$  qui permette une estimation correcte conforme à l'équation 3.1. Le MMSE vise à accomplir cela en minimisant l'erreur quadratique moyenne (*mean-squared error*; MSE), qui est donnée par l'expression suivante :

$$MSE = E \left\{ (\hat{x} - x)^2 \right\} \quad (3.3)$$

Pour minimiser cette erreur, nous tenons compte de l'importance du bruit en l'intégrant à l'équation du ZF de la façon suivante :

$$\mathbf{W} = \left( \mathbf{H}^* \mathbf{H} + \frac{\sigma_v^2}{\sigma_x^2} \mathbf{I} \right)^{-1} \mathbf{H}^* \quad (3.4)$$

Au premier coup d'œil, nous reconnaissons effectivement l'expression du ZF. À celle-ci s'est greffé le rapport entre la puissance du bruit par rapport à celle du signal  $\sigma_v^2 / \sigma_x^2$ . Nous pouvons donc observer que le ZF peut être considéré comme un cas particulier du MMSE, où le bruit aurait été nul. L'ajout de l'élément de bruit dans l'équation complexifie un peu l'algorithme, mais le garde tout de même plutôt simple, tout en améliorant les performances lorsque bien utilisé.

Toutefois, pour qu'il fonctionne à son meilleur, nous devons être en mesure d'estimer correctement l'effet du bruit. Autant le ZF n'en tient pas compte (en le mettant à 0 dans l'équation), autant lui attribuer une valeur trop élevée par rapport à la réalité nuira aux performances. Il est malheureusement souvent assez difficile de connaître la puissance exacte du bruit s'ajoutant au signal transmis au cours de son trajet, surtout dans le cas d'appareils mobiles tels que les cellulaires.

### 3.3 V-BLAST (*Vertical Bell Laboratories Layered Space-Time*)

Le V-BLAST est une technique non linéaire non et adaptative qui a été développée en 1998, suite aux études faites sur la technique BLAST de Bell Laboratories [FOS96]. À l'origine, la technique BLAST avait été développée pour exploiter au maximum la diversité des canaux inhérente à un système MIMO. Elle est aussi appelée *diagonal* BLAST (ou D-BLAST). Toutefois, bien que la technique ait donné de très bons résultats, il a été déterminé que sa complexité la rendait inappropriée à l'implantation. C'est après en être venu à cette conclusion que Bell Laboratories a développé le V-BLAST [WOL98], qui est en fait une version simplifiée du D-BLAST.

Nous allons commencer par résumer l'idée générale de l'algorithme V-BLAST. Ce dernier consiste à trouver un ordre optimal de fiabilité des informations récupérées par les antennes, et à faire une annulation successive des interférences (*successive interference cancellation*; SIC) selon l'ordre défini. Les données de l'antenne jugée comme ayant les meilleures chances d'avoir l'information correcte sont donc décodées en premier. Ensuite, l'effet causé par la série de données décodée est retiré des autres séries reçues, et ainsi de suite jusqu'à ce que l'information de la dernière antenne ait été récupérée. Nous pouvons constater que cette technique comporte tout de même un certain risque: une bonne partie

des performances dépend de la série qui est analysée en premier, puis annulée. Un mauvais choix, ou bien des erreurs dans son décodage, peuvent grandement nuire aux performances lorsque nous tentons de déterminer les données reçues par les autres antennes.

Nous pouvons trouver plusieurs techniques qui ont été proposées pour pallier à cet inconvénient du V-BLAST [BAR00], [CHO00], [SHE03], mais celles-ci ont tendance à en augmenter la complexité de façon significative. La technique de [LI04] semble plus intéressante due à sa plus faible complexité. Toutefois, nous utiliserons ici l'algorithme de base du V-BLAST. La raison justifiant ce choix est que, malgré ses nombreuses variantes, la version originale reste plus utilisée. Nous la retrouvons donc très fréquemment dans la littérature et elle sert souvent de référence pour comparer différentes techniques.

L'algorithme de détection est le suivant :

$$\tilde{\mathbf{H}} = \mathbf{H} \quad (3.5a)$$

$$\tilde{\mathbf{r}} = \mathbf{y} \quad (3.5b)$$

For  $i = 1$  to  $N_t$

$$k_i = \arg \min_{j \in \{0, \dots, m-1-i\}} \left\| \left( \tilde{\mathbf{H}}^\dagger \right)_j \right\|^2 \quad (3.5c)$$

$$\hat{y}_{k_i} = \left( \tilde{\mathbf{H}}^\dagger \right)_{k_i} \cdot \tilde{\mathbf{r}} \quad (3.5d)$$

$$\hat{s}_{k_i} = \mathbf{Q}(\hat{y}_{k_i}) \quad (3.5e)$$

$$\tilde{\mathbf{r}} = \tilde{\mathbf{r}} - \mathbf{h}_{k_i} \cdot \hat{s}_{k_i} \quad (3.5f)$$

$$\tilde{\mathbf{H}} = \tilde{\mathbf{H}}_{\bar{k}_i} \quad (3.5g)$$

End

$(\mathbf{H})_j$  représente la ligne  $j$  de la matrice  $\mathbf{H}$ , tandis que  $\mathbf{h}_j$  indique la colonne  $j$  de cette même matrice (on peut noter  $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_m]$ ). Nous allons expliquer le déroulement des opérations plus en détail dans les lignes qui suivent. Tout d'abord, nous préservons le canal ainsi que données reçues en les plaçant dans des variables qui, elles, seront modifiées en cours de route. Ensuite nous arrivons à la boucle principale de l'algorithme. En 3.5c, nous vérifions quelle ligne de la matrice du canal a la norme la moins élevée. En d'autres termes, nous trouvons celle qui est la moins affectée par les effets du canal. Le numéro de la ligne est stocké dans  $k_i$ . Ensuite, dans 3.5d, nous trouvons la valeur estimée pour cette antenne de transmission. Cette valeur est démodulée selon la constellation en 3.5e (la fonction de démodulation étant représentée par  $\mathbf{Q}$ ). L'étape qui suit est celle où se fait l'annulation de l'effet de l'antenne traitée. Nous soustrayons du signal reçu l'information maintenant connue de l'antenne qui a été estimée (incluant l'effet du canal) en 3.5f. Enfin, la matrice de canal peut être réduite en enlevant la colonne correspondant à l'antenne de transmission annulée (3.5g). La boucle se répète jusqu'à ce que toutes les valeurs aient été estimées. À chaque itération, une antenne est ainsi évaluée, puis son effet est enlevé du signal reçu pour faciliter les décodages subséquents, et ainsi améliorer leur performance.

La technique que nous avons ici présentée est en fait celle du ZF-VBLAST. En effet, tout comme pour le ZF, nous pouvons observer qu'en aucun cas l'algorithme ne tient compte du bruit. C'est pourquoi nous pouvons l'améliorer en appliquant la technique MMSE, ce qui donne le MMSE-VBLAST. Le passage de ZF-VBLAST à MMSE-VBLAST est fort simple au niveau algorithmique. En 3.5, il suffit de remplacer la pseudo-inverse  $\mathbf{H}^\dagger$  par l'expression suivante :

$$\mathbf{H}_{MMSE}^\dagger = \left( \mathbf{H}^* \mathbf{H} + \frac{\sigma_v^2}{\sigma_x^2} \mathbf{I} \right)^{-1} \mathbf{H}^* \quad (3.6)$$

soit la définition même de la technique MMSE discutée à la section précédente. Évidemment, l'augmentation au niveau des calculs est du même ordre que lorsque nous passons de ZF à MMSE.

### 3.4 Least mean-square (LMS)

Contrairement aux techniques précédentes, le LMS est une technique adaptative linéaire. Le LMS, découvert en 1960, tire ses origines d'un algorithme appelé *gradient descent* (« descente de gradient »). C'est l'algorithme adaptatif le plus largement utilisé dans plusieurs domaines, tels que le filtrage, l'identification de systèmes et l'égalisation. Nous ne référons toutefois pas le développement qui y a conduit, puisque cela a été fait à maintes reprises et que plusieurs sont déjà familiers avec. Voici donc le trio d'équations du LMS [HAY96] :

$$\mathbf{x}_{est} = \mathbf{W}_i \mathbf{y}_i \quad (3.7)$$

$$\mathbf{e} = \mathbf{x}_{des} - \mathbf{x}_{est} \quad (3.8)$$

$$\mathbf{W}_{i+1} = \mathbf{W}_i + \mu \mathbf{e} \mathbf{y}_i^* \quad (3.9)$$

$\mathbf{W}_i$  est une matrice de  $N_t$  lignes par  $N_r$  colonnes contenant les poids à l'itération  $i$  ( $i$ -ème groupe de données d'adaptation reçu);  $\mathbf{e}$  est l'erreur calculée entre les données estimées et celles désirées;  $\mu$  est le pas d'adaptation, que nous nous devons d'ajuster pour un bon fonctionnement (généralement une faible valeur positive près de zéro).

Les équations ayant été énoncées, nous pouvons expliquer quelques notions importantes relatives au fonctionnement du LMS. Tout d'abord, l'adaptation elle-même, soit la mise à jour des poids en 3.9, ne se fait que lors de l'envoi de données d'adaptation. Celles-ci sont connues à la fois du transmetteur et du récepteur, et ont une forme identique à ce à quoi pourrait ressembler une série de données normale. Elles consistent généralement en une série d'apparence aléatoire, de façon à couvrir un maximum de possibilités de données de transmission. Plus les données d'adaptation ressemblent à de « vraies » données, meilleure sera l'adaptation. Une fois l'adaptation faite, et ce, jusqu'à la prochaine série d'adaptation, seule l'équation 3.7 est utilisée pour estimer les données, avec les poids préalablement calculés.

Aussi, nous devons observer que même si le canal n'intervient pas directement dans les équations, il est évidemment compensé. De la même façon, le bruit est aussi exclu des équations tout en étant compensé lors de l'égalisation, contrairement au ZF. Comme nous l'avons mentionné précédemment, l'un des intérêts d'utiliser une technique adaptative est qu'elle ne nécessite pas d'estimateur de canal. En fait, la technique ne fera jamais la distinction entre les interférences dues au canal par rapport à celles du bruit. Mais quelles que soient leurs sources, tant qu'elles restent à peu près semblables à ce qu'elles étaient au moment de l'adaptation, les interférences sont prises en compte et leur effet est limité par le LMS. Ce comportement est caractéristique des techniques adaptatives.

Toutefois, le LMS a le défaut de prendre un certain temps à trouver les poids à utiliser. Cela force donc à utiliser une série d'adaptation plus longue au prix d'un peu de vitesse de transfert des données d'information. Un peu comme le ZF était la technique non adaptative d'égalisation la plus simple, le LMS est la plus simple des techniques adaptatives. Il est



donc prévisible que sa simplicité limite ses performances générales. De plus, le pas d'adaptation joue un rôle important dans l'estimation. Un pas mal ajusté retardera la convergence des poids, et peut même l'empêcher complètement.

### 3.5 Recursive least-squares (RLS)

Le RLS est un autre algorithme linéaire adaptatif largement utilisé. D'une complexité plus grande, celui-ci offre en contrepartie une convergence beaucoup plus rapide. Nous retrouvons donc toujours cette dualité entre l'efficacité et la complexité. Le raisonnement qui conduit à cette technique a été présenté dans maints ouvrages de référence, dont [HAY96].

Avant de plonger dans les équations, mentionnons que le RLS est utilisé de la même façon que le LMS (et que n'importe quel autre égaliseur adaptatif); cela signifie que l'adaptation des poids est faite sur une séquence connue, pour ensuite évaluer les véritables données avec les poids calculés. Voici l'algorithme du RLS :

$$\text{Initialisation : } \mathbf{P}_0 = \pi_0 \mathbf{I} \quad (3.10)$$

$$\mathbf{x}_{est} = \mathbf{W}_i \mathbf{y}_i \quad (3.11)$$

$$\mathbf{e} = \mathbf{x}_{des} - \mathbf{x}_{est} \quad (3.12)$$

$$\mathbf{P}_{i+1} = \frac{1}{\lambda} \left( \mathbf{P}_i - \frac{\lambda^{-1} \mathbf{P}_i \mathbf{y}_i \mathbf{y}_i^* \mathbf{P}_i}{1 + \lambda^{-1} \mathbf{y}_i^* \mathbf{P}_i \mathbf{y}_i} \right) \quad (3.13)$$

$$\mathbf{W}_{i+1} = \mathbf{W}_i + \mathbf{e} \mathbf{y}_i^* \mathbf{P}_{i+1} \quad (3.14)$$

Comme nous le voyons dans l'équation 3.10, la matrice  $\mathbf{P}$  est initialisée par un scalaire (généralement de valeur la plus élevée possible) multiplié à une matrice identité de taille

$N_r \times N_r$ . Cette phase d'initialisation faite, nous estimons ensuite le vecteur d'informations transmises à l'aide des poids calculés jusqu'à maintenant (0 à la toute première estimation) en 3.11. Suite à cela, l'erreur est calculée dans l'équation 3.12 par rapport aux données désirées. Puis la matrice  $\mathbf{P}$  est mise à jour en 3.13. Pour ce faire, nous faisons usage du facteur d'oubli  $\lambda$ , qui prend généralement une valeur près de 1. Ce facteur d'oubli dicte la quantité de valeurs passées prises en compte dans les calculs de coefficients (avec 1, toutes les valeurs sont prises en compte). Enfin, l'équation 3.14 calcule les nouvelles valeurs de la matrice de poids  $\mathbf{W}$ . Comme pour le LMS, une fois l'adaptation terminée, seule l'équation 3.11 est utilisée pour faire l'égalisation avec les poids obtenus lors de l'adaptation.

Par définition, le RLS minimise l'erreur quadratique. En contraste, le LMS minimise l'erreur quadratique moyenne. Ce dernier dépend donc des statistiques du signal, tandis que le RLS dépend directement du signal lui-même. C'est ce qui explique en bonne partie la plus grande précision qui est attribuée au RLS dans sa recherche de poids.

Nous devons faire attention au facteur d'oubli en utilisant le RLS, car un peu comme pour le pas d'adaptation du LMS, les résultats peuvent varier beaucoup selon la valeur qui lui est attribuée. De plus, notons qu'une valeur de 1 utilise toutes les données passées. À l'opposé, en réduisant cette valeur, nous réduisons la fenêtre de données utiles pour calculer les poids. En général, nous voulons être près de 1, mais sans y être, car le fait d'utiliser toutes les données vient à minimiser l'importance des données récentes. La conséquence à cela est donc une perte en adaptabilité, surtout si les conditions sont changeantes.

Malgré le fait que les équations sont plus lourdes que pour le LMS, le fait que le RLS ne contienne aucune inversion de matrice fait qu'il n'est tout de même pas très difficile à implémenter.

### 3.6 Zero-Forcing Sorted *QR* decomposition (ZF-SQRD)

La technique *Sorted QR decomposition* (« Décomposition QR ordonnée »; SQRD) est moins populaire que la plupart des autres mentionnées. Cela a compliqué un peu la recherche d'informations, mais celles trouvées ont permis de la reproduire [WUB01], [WUB03], et même de l'ajuster à nos besoins, car selon les articles consultés, elle ne semble pas avoir été utilisée pour des systèmes MIMO-OFDM jusqu'à présent. Il s'agit d'une technique linéaire non adaptative basée sur le V-BLAST. Nous débutons par la version ZF, la version MMSE étant présentée à la section suivante, car elle comporte quelques différences importantes.

Comme nous en avons discuté, le V-BLAST est populaire car il réussit à bien exploiter la diversité des canaux. Toutefois, de nombreuses inversions de matrices sont nécessaires pour y arriver. L'inversion de matrice est l'un des calculs les plus difficiles à implémenter, car chaque inversion demande de nombreuses multiplications, dont le nombre augmente rapidement avec la taille des matrices. La matrice la plus souvent inversée est celle du canal, dont la taille est définie par le nombre d'antennes de transmission et de réception, ce qui fait que plus leur nombre augmente, plus les inversions de matrices sont lourdes au niveau calcul.

Là où le SQRD trouve sa raison d'être est dans le fait qu'il s'agit d'un algorithme sans inversion de matrice. Rappelons que la définition générale d'une inversion de matrice est la suivante :

$$\mathbf{X}^{-1} = \frac{1}{\det(\mathbf{X})} \text{Adj}(\mathbf{X}) \quad (3.15)$$

Le déterminant et la matrice de cofacteurs demandent chacun de nombreuses opérations pour être calculés, et ce nombre augmente exponentiellement en fonction de la taille de la matrice en question. De surcroît, comme nous l'avons expliqué plus tôt, l'inversion que nous utilisons est celle de la pseudo-inverse de Moore-Penrose, décrite par l'équation 3.6, ce qui ajoute encore des opérations à une quantité déjà importante. Il est donc facile de comprendre pourquoi nous tentons de limiter au maximum l'usage de cette opération.

Le but du SQRD est donc d'éviter l'opération matricielle la plus imposante afin d'alléger les calculs, tout en conservant des performances se rapprochant le plus possible de celles du V-BLAST, sur lequel la technique est basée. Contrairement à toutes les autres techniques proposées jusqu'ici, la recherche bibliographique effectuée ne nous a pas permis de la trouver appliquée au MIMO-OFDM. Il sera donc particulièrement intéressant de l'adapter et de la comparer au V-BLAST sur notre plate-forme de simulation. Comme cette technique est largement moins documentée que les précédentes, nous allons l'expliquer plus en détail.

La technique SQRD est basée sur le ZF-VBLAST, et nous faisons usage de l'équation 2.1, que nous réécrivons pour mieux expliquer les origines du SQRD :

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}$$

Dans cette équation, nous pouvons exprimer la matrice de canal  $\mathbf{H}$  par sa décomposition QR. L'opération de décomposition QR factorise une matrice (dans notre cas, celle du canal) en un produit de deux matrices.

$$\mathbf{H} = \mathbf{Q}\mathbf{R} \tag{3.16}$$

Dans une décomposition QR d'une matrice de dimensions  $N_r \times N_t$ ,  $\mathbf{Q}$  est une matrice unitaire de taille  $N_r \times N_r$ , et  $\mathbf{R}$  en est une triangulaire supérieure. L'équation 2.1 devient :

$$\mathbf{y} = \mathbf{Q}\mathbf{R}\mathbf{x} + \mathbf{v} \quad (3.17)$$

En multipliant les deux côtés de cette équation par  $\mathbf{Q}^*$ , nous obtenons, grâce aux propriétés des matrices mentionnées plus haut :

$$\mathbf{y}' = \mathbf{Q}^*\mathbf{y} = \mathbf{R}\mathbf{x} + \boldsymbol{\eta} \quad (3.18)$$

le terme  $\boldsymbol{\eta}$  résultant du produit de  $\mathbf{Q}^*$  et du bruit  $\mathbf{v}$  garde les propriétés statistiques de  $\mathbf{v}$  intactes, puisque  $\mathbf{Q}^*$  est unitaire.  $\mathbf{y}'$  représente simplement un vecteur modifié de données reçues. Un élément  $k$  du vecteur  $\mathbf{y}'$  peut donc être exprimé par :

$$y'_k = r_{k,k}x_k + \eta_k + d_k \quad (3.19)$$

Dans cette équation, le terme  $d_k$  représente en fait un terme d'interférence défini équivalent à :

$$d_k = \sum_{j=k+1}^{N_t} r_{k,j}x_j \quad (3.20)$$

Toutefois, parce que  $\mathbf{R}$  est une matrice triangulaire supérieure,  $d_k$  est indépendant de  $x_1, \dots, x_{k-1}$ . Cela permet d'identifier le terme  $\hat{x}_{N_t} = Q[y'_{N_t}/r_{N_t,N_t}]$ , où  $Q$  représente la quantification appropriée selon la constellation que nous avons choisie. Ensuite, pour  $1 \leq k \leq N_t$ , le terme d'interférence peut être annulé en assumant que les valeurs  $\hat{x}_{k+1}, \dots, \hat{x}_{N_t}$  sont correctes. Nous pouvons donc réécrire 3.19 de cette façon :

$$z_k = y'_k - d_k = r_{k,k}x_k + \eta_k \quad (3.21)$$

et utiliser ce  $z_k$  pour trouver :

$$\hat{x}_k = Q[y'_k / r_k] \quad (3.22)$$

L'ordre de détection est très important dans l'efficacité d'une telle technique. Un  $\mathbf{R}$  optimal assure un rapport signal sur bruit (*signal-to-noise-ratio*; SNR) maximal à chaque étape de détection. L'ordre s'établit généralement en interchangeant des colonnes de  $\mathbf{H}$  ainsi que des rangées de  $\mathbf{x}$ . Une façon de déterminer le meilleur ordre serait de faire de nombreuses permutations de  $\mathbf{H}$ , et de refaire chaque fois la décomposition QR jusqu'à avoir un  $\mathbf{R}$  optimal. Toutefois, il ne s'agit pas d'une technique efficace au niveau des calculs à faire.

Plutôt que d'itérer avec différentes matrices de canal jusqu'à trouver le meilleur ordre, nous utiliserons un algorithme connu, celui de Gram Schmidt, pour calculer la matrice  $\mathbf{R}$  ligne par ligne de haut en bas, tout en ordonnant la matrice  $\mathbf{Q}$  colonne par colonne de gauche à droite. Le but étant de faire une seule décomposition QR, l'algorithme propose de trouver le  $\mathbf{H}$  qui minimise chaque  $|r_{k,k}|$ , pour les valeurs de  $k$  allant de 1 à  $N_t$  en laissant inchangées les valeurs  $r_{j,j}$  pour  $j < k$ . Voici l'algorithme en question :

$$\mathbf{R} = 0, \mathbf{Q} = \mathbf{H}, \mathbf{p} = (1, \dots, N_t) \quad (3.23a)$$

For  $i = 1, \dots, N_t$

$$k_i = \arg \min_{j=i, \dots, N_t} |\mathbf{q}_j|^2 \quad (3.23b)$$

$$\text{Échanger les colonnes } i \text{ et } k_i \text{ dans } \mathbf{Q}, \mathbf{R} \text{ et } \mathbf{p}. \quad (3.23c)$$

$$r_{i,i} = |\mathbf{q}_i| \quad (3.23d)$$

$$\mathbf{q}_i = \mathbf{q}_i / r_{i,i} \quad (3.23e)$$

For  $j = i+1, \dots, N_t$

$$r_{i,j} = \mathbf{q}_i^* \mathbf{q}_j \quad (3.23f)$$

$$\mathbf{q}_j = \mathbf{q}_j - r_{i,j} \mathbf{q}_i \quad (3.23g)$$

End

End

La matrice  $\mathbf{Q}$  est donc initialisée à l'aide de celle du canal, tandis que  $\mathbf{R}$  débute à 0, et qu'un vecteur d'indexage est créé. Rappelons que  $\mathbf{q}_j$  signifie la  $j$ -ème colonne de la matrice  $\mathbf{Q}$ . En 3.23b, nous débutons sensiblement de la même façon que le V-BLAST le faisait (3.5c). Ensuite, nous faisons la permutation, qui a pour effet de placer en premier les colonnes qui ont normalement le meilleur SNR, jusqu'aux moins bonnes à la fin. Après, les étapes suivantes font le remplissage de  $\mathbf{R}$ , tout en ajustant  $\mathbf{Q}$ , afin de garder valide l'équivalence de l'équation 3.16. Comme nous le voyons, il s'agit strictement d'un algorithme pour réordonner les colonnes de  $\mathbf{Q}$  et  $\mathbf{R}$ . Une fois ces matrices trouvées, les équations 3.18 à 3.22 sont utilisées pour estimer les données. Les performances attendues devraient se rapprocher de celles du ZF-VBLAST.

Comme précédemment, nous avons présenté la version ZF de la technique, qui ne tient pas compte de la puissance du bruit. Nous présentons la version MMSE dans la prochaine section, car cette fois-ci, pour passer de ZF à MMSE, il faut faire plus que changer une seule équation.

### 3.7 Minimum Mean-Squared Error Sorted QR decomposition (MMSE-SQRD)

Le MMSE-SQRD est basé sur l'observation du MMSE de base. En effet, il a été observé que les matrices de l'équation 2.1 peuvent être écrites différemment. Nous appellerons ces matrices les matrices étendues :

$$\underline{\mathbf{H}} = \begin{bmatrix} \mathbf{H} \\ \sigma_n \mathbf{I}_{N_r} \end{bmatrix} \quad \text{et} \quad \underline{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ 0_{N_r,1} \end{bmatrix} \quad (3.24)$$

où  $\sigma_n = \sigma_v / \sigma_x$ , soit le rapport du bruit sur le signal. Avec ces matrices, nous pouvons réécrire l'équation 3.2 du ZF presque intacte :

$$\mathbf{W} = \underline{\mathbf{H}}^\dagger = (\underline{\mathbf{H}}^\star \underline{\mathbf{H}})^{-1} \underline{\mathbf{H}}^\star \quad (3.25)$$

Nous avons donc une équation de la forme du ZF, mais dont les matrices tiennent compte du bruit. C'est de cette observation que découle la technique MMSE-SQRD. En effet, l'équation du MMSE ayant pris la même forme que celle du ZF, une démarche très semblable peut maintenant être utilisée pour ordonner les matrices et faire l'égalisation du signal. L'algorithme en question est le suivant :

$$\underline{\mathbf{R}} = 0, \quad \underline{\mathbf{Q}} = \underline{\mathbf{H}}, \quad \mathbf{p} = (1, \dots, N_r) \quad (3.26a)$$

For  $i = 1, \dots, N_r$

$$\mathbf{norm}_i = |\mathbf{q}_i|^2 \quad (3.26b)$$

End

For  $i = 1, \dots, N_r$



$$k_i = \arg \min_{j=i, \dots, N_i} \mathbf{norm}_j \quad (3.26c)$$

Échanger les colonnes  $i$  et  $k_i$  dans  $\mathbf{R}$  et  $\mathbf{p}$ ,  $\mathbf{norm}$ , et les  $N_r + i - 1$

premières rangées de  $\mathbf{Q}$  (3.26d)

$$r_{i,i} = \sqrt{\mathbf{norm}_i} \quad (3.26e)$$

$$\mathbf{q}_i = \mathbf{q}_i / r_{i,i} \quad (3.26f)$$

For  $j = i + 1, \dots, N_i$

$$r_{i,j} = \mathbf{q}_i^* \mathbf{q}_j \quad (3.26g)$$

$$\mathbf{q}_j = \mathbf{q}_j - r_{i,j} \mathbf{q}_i \quad (3.26h)$$

$$\mathbf{norm}_j = \mathbf{norm}_j - |r_{i,j}|^2 \quad (3.26i)$$

End

End

La seule différence que nous remarquons au premier coup d'œil dans l'algorithme de classement, c'est la présence de  $\mathbf{norm}$ . En effet, la norme des colonnes de  $\mathbf{Q}$  est calculée une fois au début, puis mise à jour lorsqu'il y a changement. Cela aurait certainement pu être fait dans le ZF-SQRD, mais ça montre une progression au niveau de l'optimisation de l'algorithme entre les articles. Il faut noter qu'une erreur s'était glissée dans l'équation 3.26i dans l'article de référence [WUB03]; elle est ici corrigée, car elle empêchait l'algorithme de fonctionner correctement.

Cette technique sera comparée principalement au MMSE-VBLAST, car elle est supposée d'en approcher les performances, malgré la diminution non négligeable de la charge de calcul.

### 3.8 *Simplified sphere decoding (SSD)*

Avant d'aborder le SSD, nous allons glisser quelques mots sur ses origines. Comme le nom l'indique, il s'agit d'une version simplifiée de l'algorithme de *sphere decoding* (SD). Cette dernière est une technique alternative au *maximum likelihood* (ML), qui offre des résultats optimaux au prix d'un énorme effort de calcul.

En fait, le ML est très peu utilisé en pratique, car la lourdeur de ses calculs le rend impossible à implémenter dans la plupart des applications. Nous allons en effleurer le concept, afin de pouvoir poursuivre et justifier le SD, puis le SSD. Le ML repose sur le concept de treillis (*lattice*). La constellation QAM que nous utilisons est un treillis (ensemble de points sur les axes réels-imaginaires; voir Figure 3), tandis que cette même constellation affectée par le canal en donne un autre, une version déformée de celui d'origine. Le signal reçu apparaît donc dans ce deuxième treillis après avoir traversé le canal. C'est à partir de ce moment que le ML, le SD et le SSD diffèrent.

Le ML vise à réduire l'expression  $\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2$ . Pour ce faire, il calculera la distance du point reçu avec chacun des points de la constellation individuellement. Si nous fonctionnons en QAM-64, cela fait 64 points avec lesquels comparer la distance. Nous pouvons facilement voir que le problème d'estimation prend des proportions impressionnantes avec une telle technique.

Le SD vise à réduire la même expression que le ML. Toutefois, il restreint sa recherche à une sphère de rayon  $\sqrt{C}$  centrée sur le point à décoder, ce qui lui donne son nom. Contrairement aux techniques précédentes, les techniques basées sur les treillis ne fonctionnent qu'avec des valeurs réelles. Il nous faudra réécrire l'équation 2.1 en utilisant des matrices réelles. Nous utiliserons les matrices suivantes :

$$\bar{\mathbf{y}} = \begin{bmatrix} \text{Re}(\mathbf{y}^T) & \text{Im}(\mathbf{y}^T) \end{bmatrix}^T \quad (3.27)$$

$$\bar{\mathbf{x}} = \begin{bmatrix} \text{Re}(\mathbf{x}^T) & \text{Im}(\mathbf{x}^T) \end{bmatrix}^T \quad (3.28)$$

$$\bar{\mathbf{H}} = \begin{bmatrix} \text{Re}(\mathbf{H}) & -\text{Im}(\mathbf{H}) \\ \text{Im}(\mathbf{H}) & \text{Re}(\mathbf{H}) \end{bmatrix} \quad (3.29)$$

$$\bar{\mathbf{v}} = \begin{bmatrix} \text{Re}(\mathbf{v}^T) & \text{Im}(\mathbf{v}^T) \end{bmatrix}^T \quad (3.30)$$

Avec ces matrices, nous pouvons utiliser l'algorithme du SD illustré à la figure 8, tirée de [CHA02].

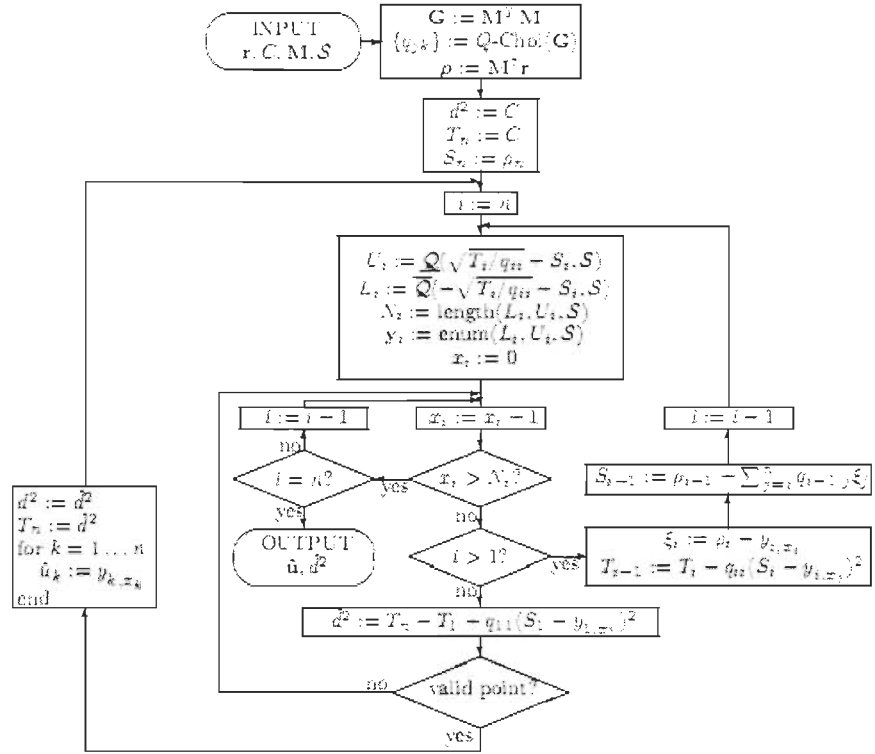


Figure 8 : Algorithme du SD

Commençons par définir la notation utilisée. Tout d'abord, les valeurs d'entrées de l'algorithme  $\mathbf{r}$ ,  $C$ ,  $\mathbf{M}$  sont respectivement celles que nous avons définies comme étant  $\mathbf{y}$ , le carré du rayon de la sphère et  $\bar{\mathbf{H}}$ .  $S$  représente quant à lui les données possibles du QAM. Par exemple, pour un 4-QAM dont les valeurs possibles sont  $[-1,-1]$ ,  $[-1,1]$ ,  $[1,-1]$ ,  $[1,1]$ , nous aurions  $S_4 = \{-1,1\}$ , soit les deux seules valeurs que nous retrouvons dans les 4 combinaisons du 4-QAM défini.  $Q$ -Chol est une combinaison de deux opérations. Tout d'abord, nous faisons une factorisation de Cholesky sur la matrice  $\mathbf{G}$  pour récupérer une matrice triangulaire supérieure  $\mathbf{R}$ . Ensuite, les valeurs de  $\mathbf{Q}$  sont calculées comme suit :  $q_{j,j} = r_{j,j}^2$  pour  $j = 1, \dots, n$  ( $n$  étant la taille de la matrice, soit  $2N_t$ ),  $q_{j,k} = r_{j,k} / r_{j,j}$  pour  $j = 1, \dots, n$  et  $k = j+1, \dots, n$ . La fonction  $\underline{Q}(v, S)$  arrondit l'élément  $v$  à la valeur inférieure

la plus près dans l'ensemble  $S$  ( $-\infty$  s'il n'y en a pas). De même,  $\bar{Q}(v, S)$  arrondit  $v$  vers l'élément supérieur de  $S$  le plus près ( $\infty$  s'il n'y en a pas). Par exemple,  $\bar{Q}(v=1.2, S_{16}=\{-3, -1, 1, 3\})$  donne une valeur de 3. La fonction  $\text{enum}(L_i, U_i, S)$  liste toutes les valeurs de  $S$  entre les bornes  $L_i$  et  $U_i$  inclusivement dans un vecteur en rangée, tandis que la fonction  $\text{length}$  donne le nombre de ces éléments. La sortie de cet algorithme est notée  $\hat{\mathbf{u}}$  sur la figure 8, qui est la valeur estimée trouvée.

Nous allons poursuivre avec un exemple simple pour voir comment l'algorithme procède pour estimer sa valeur, et aussi pour comparer avec l'algorithme SSD qui sera présenté. Gardons la notation de la figure 8 pour éviter toute confusion ( $\mathbf{u}$  pour le signal transmis,  $\mathbf{r}$  pour celui reçu,  $\mathbf{M}$  pour le canal) :

$$\mathbf{r} = [4.08 \quad -0.81]^T \quad (3.31)$$

$$\mathbf{u} = [1 \quad -3]^T \quad (3.32)$$

$$\mathbf{M} = \begin{bmatrix} 0.5 & -1 \\ 1 & 0.5 \end{bmatrix} \quad (3.33)$$

$$\mathbf{v} = [0.58 \quad -0.31]^T \quad (3.34)$$

Le système a une seule antenne de transmission et de réception. Fixons  $C$  à 49. La constellation utilisée est un 64-QAM dont  $S_{64} = \{-7, -5, -3, -1, 1, 3, 5, 7\}$ . Les coordonnées de  $\mathbf{r}$  par rapport au treillis sont  $\boldsymbol{\rho} = \mathbf{M}^{-1}\mathbf{r} = [0.984 \quad -3.588]^T$ . L'algorithme commence par la seconde coordonnée. La recherche de valeurs probables pour celle-ci donne  $\mathbf{y}_2 = [-7 \quad -5 \quad -3 \quad -1 \quad 1]$ . Commençant avec la première valeur, soit  $u_2 = -7$ , le second

passage dans le bloc où se fait l'énumération donne des valeurs possibles  $y_1 = [-3 \ -1 \ 1 \ 3 \ 5]$  pour la coordonnée  $u_1$ . Maintenant que des possibilités ont été trouvées pour chaque coordonnée, la recherche de points peut commencer. L'algorithme débute avec la première valeur de  $y_2$ , -7, parcourt les trois premiers points qu'il rencontre dans l'ordre, soit  $[-3 \ -7]^T$ ,  $[-1 \ -7]^T$  et  $[1 \ -7]^T$ . À chaque fois, le rayon de la sphère est réduit pour s'ajuster au nouveau point, ce qui indique que l'algorithme s'approche du point recherché. Au quatrième point,  $[3 \ -7]^T$  la sphère veut s'agrandir, ce qui signifie que nous nous éloignons du centre. L'algorithme passe alors à la valeur suivante de  $y_2$ , -5, et répète la procédure, avec le cercle réduit. Le tout est répété jusqu'à ce qu'il n'y ait plus de point en mesure de réduire le rayon du cercle, ce qui signifie que nous avons trouvé le point le plus rapproché, qui constitue l'estimation du détecteur ( $[1 \ -3]^T$  dans ce cas).

Nous constatons que même pour un exemple aussi simple, la procédure est assez longue. Au total, l'algorithme aura parcouru six points dans sa recherche de la solution, tel qu'illustré sur la figure 9 :

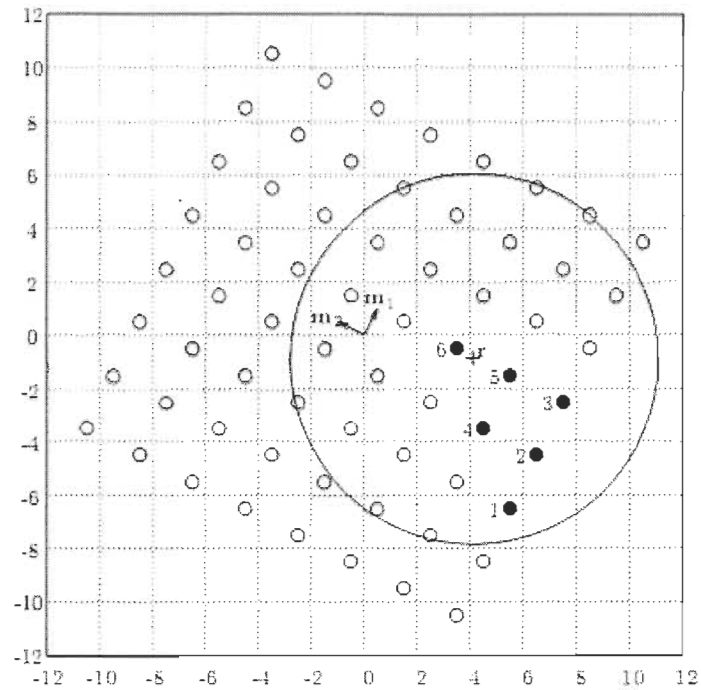


Figure 9 : Exemple de recherche de point du SD montrant les points inspectés

Il est intéressant de noter que la taille du cercle est primordiale dans quantité de calculs à faire. Un cercle plus petit aurait amélioré la vitesse en réduisant le nombre de points à rechercher. Toutefois, il n'est pas facile de bien déterminer la taille idéale, donc nous devons souvent le mettre plus grand pour être certain de trouver des points. Les performances du SD sont, avec un cercle permettant de trouver toutes les valeurs, les mêmes que celles du ML.

Le SSD qui nous intéresse ici ressemble beaucoup au SD dont il est inspiré dans sa forme. La grande faiblesse du SD, comme nous avons pu le constater, est la façon dont il zigzague en partant du pourtour du cercle en se dirigeant vers le centre. C'est donc là où le SSD offre du changement : il tente de débiter sa recherche le plus près possible du centre du cercle, pour ainsi la terminer beaucoup plus rapidement. L'algorithme du SSD [CHA02] est illustré à la figure 10.

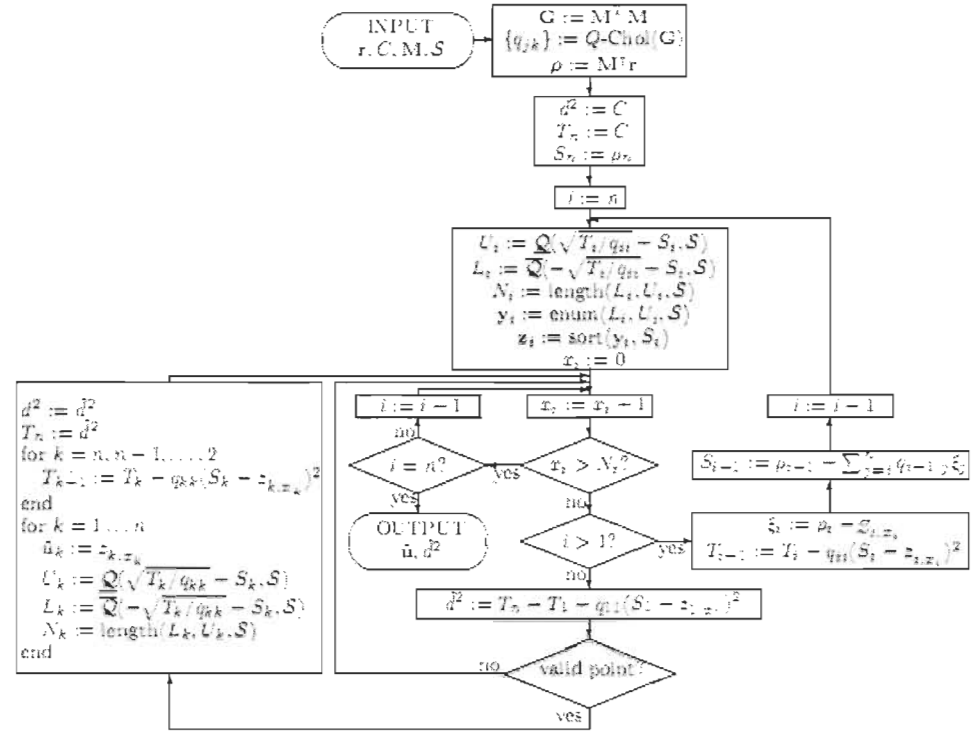


Figure 10 : Algorithme du SSD

La principale différence du SSD est son utilisation d'un algorithme qui classe en ordre les coordonnées possibles en commençant par les plus près du centre de l'intervalle défini par  $L_i$  et  $U_i$ . Le critère de classement est le suivant :

$$|y_{i,j} - S_i|^2 \quad (3.35)$$

Reprenons notre exemple précédent pour illustrer le changement. Au premier passage, l'algorithme viendra à la conclusion que les valeurs possibles pour la seconde coordonnée pour sont  $\mathbf{z}_2 = [-3 \ -5 \ -1 \ -7 \ 1]$  ( $\mathbf{z}$  correspond au  $\mathbf{y}$  une fois classé). Ensuite, en commençant avec  $u_2 = -3$ , pour la première coordonnée, il trouvera les possibilités  $\mathbf{z}_1[1 \ -1 \ 3 \ -3 \ 5 \ -5 \ 7]$ . Le premier et seul point testé sera donc  $[1 \ -3]$ , car aucun



autre point ne peut réduire le cercle davantage. La valeur aura été décodée dès le premier essai, contrairement aux six du SD.

Remarquons qu'une erreur présente sur l'algorithme dans la référence a été corrigée ici. En effet, le  $z$  rouge remplace un  $y$  sur la figure 10, qui, visiblement n'avait pas sa place à cet endroit. Le  $z$  est la version ordonnée du  $y$ , qui n'est pas utilisé dans cette version simplifiée. Cette erreur rendait les résultats instables selon les circonstances. Dans un autre ordre d'idées, notons que le « simplifié » du nom de l'algorithme est plutôt une optimisation des calculs qu'une simplification. L'algorithme fonctionne plus efficacement, tout en donnant des résultats parfaitement identiques à celui du SD. Il ne sacrifie donc pas les performances au profit de la vitesse. Rappelons que cette technique ne semble pas, selon la littérature, avoir été utilisée avec des systèmes MIMO-OFDM.

Ceci termine le tour d'horizon des techniques qui seront comparées sur la plate-forme de simulation. Nous aurons remarqué que certaines techniques ont bénéficié d'explications plus détaillées, dans le but de tenter de compenser la moins grande quantité de documentation à leur sujet. Nous allons poursuivre au prochain chapitre en décrivant la plate-forme dans laquelle prendront place ces détecteurs.

## Chapitre 4 - Simulations et résultats

Dans ce chapitre, nous allons parler des simulations qui ont été faites sur la plate-forme, et analyser les résultats obtenus. Tout d'abord, commençons par expliquer en quoi consiste la plate-forme en question.

### 4.1 Plate-forme de simulation

La plate-forme utilisée est une version améliorée d'une plate-forme ayant déjà été faite à l'Université du Québec à Trois-Rivières par M. Adel Omar Dahmane, elle-même basée sur celle présentée dans [WAN04]. Plusieurs éléments ont été ajoutés, afin de répondre à nos besoins, qui sont d'avoir une plate-forme capable de prendre en charge plusieurs algorithmes d'égalisation de formes très diverses, et d'en simuler le comportement dans des conditions définies par l'utilisateur.

La plate-forme suit donc le parcours du signal tel que nous l'avions illustré aux Figures 6 et 7. Avant même d'avoir un signal à envoyer, nous devons procéder à une étape d'initialisation. C'est ici que sont définis tous les paramètres qui définissent le système MIMO-OFDM qui sera utilisé lors de la simulation. Nous y choisissons le nombre d'antennes de transmission et de réception à utiliser. La plage de SNR est définie. Comme les graphiques que nous obtiendrons illustreront le taux d'erreur sur les bits (*bit error rate*; BER), cette plage dictera les SNR pour lesquels le BER sera calculé. Ensuite, une variété de paramètres permettent de configurer la modulation OFDM à souhait. Nous pouvons y

définir le nombre de porteuses à utiliser, leur contenu (certaines peuvent être réservées pour des pilotes par exemple), ainsi que le nombre de paquets de symboles OFDM, avec le nombre de symboles par paquet, etc.

Ensuite vient l'étape de la transmission. C'est ici que nous créons le signal à envoyer en fonction de ce que nous avons spécifié à l'initialisation. Les symboles sont répartis également entre les antennes de transmission. Ces symboles ont des valeurs possibles de  $[0, \dots, N_{QAM} - 1]$  (par exemple, 0 à 15 pour du 16-QAM). Ils sont ensuite modulés selon la constellation QAM définie lors de l'initialisation, devenant des nombres complexes. S'il y a des pilotes à ajouter, c'est ici qu'ils le seront, tout comme le CP (il peut être ajouté à n'importe quel moment, mais c'est ici que ça se fait dans la plate-forme). Vient ensuite l'IFFT, qui est appliquée au signal pour le faire passer du domaine fréquentiel au domaine temporel. Le signal est prêt à être envoyé pour la suite de la simulation. En pratique, c'est ici que la modulation fait en sorte que les fréquences des sous-porteuses soient orthogonales. Toutefois, la plate-forme ne s'étend pas à l'aspect analogique du signal, et considère que cette opération se fait de sorte que les fréquences sont parfaitement orthogonales, c'est-à-dire, sans aucune ICI. Bien que ça ne soit pas utilisé dans les simulations que nous ferons, mentionnons aussi la possibilité d'ajouter une non-linéarité de type saturation juste avant l'envoi, pour simuler les amplificateurs en saturation.

Après la transmission, la plate-forme simule la partie où le signal voyage jusqu'à sa destination : le canal. En chemin, la multiplication avec les coefficients de la matrice du canal se fait, et un bruit blanc gaussien est ajouté (d'une importance dépendant du SNR déterminé lors de l'initialisation).

Enfin vient l'étape de la réception. La première étape de la réception en est une que nous pourrions appeler de filtrage des données, car elle consiste à retirer les données qui ne sont pas des informations (pilotes, CP). Les données sont ensuite ramenées au domaine fréquentiel par l'opération de FFT. L'étape suivante est celle de l'estimation de canal. Comme nous l'avons mentionné précédemment, cette étape n'est utile qu'avec les techniques d'égalisation non adaptatives, car les techniques adaptatives n'ont aucunement besoin de connaître les valeurs du canal pour effectuer leur égalisation. Toutefois, pour faire une étude comparative des performances des égaliseurs, nous considérons une connaissance parfaite du canal, donc nous n'utiliserons pas d'estimateur. À l'opposé, l'étape qui suit est celle qui nous intéresse vraiment : celle de l'égalisation. C'est à cet endroit que se fait l'égalisation selon la technique spécifiée dans l'initialisation. Selon la technique, nous devons poursuivre ou non avec une quantification selon la constellation utilisée. Certaines techniques intègrent déjà cette étape à leur fonctionnement (V-BLAST par exemple), alors que d'autres non (comme le ZF). Enfin, les valeurs estimées sont comparées à celles générées au départ lors de la transmission pour calculer le BER.

Maintenant que la structure générale de la plate-forme a été expliquée, nous pouvons passer aux diverses simulations. Celles-ci montreront le comportement de diverses techniques, dans des conditions parfaitement identiques. À moins que le contraire ne soit mentionné, les simulations sont faites pour un système MIMO-OFDM à 64 sous-porteuses, dont 48 contiennent des données (les autres étant surtout des pilotes, qui n'ont pas d'impact sur ces simulations). La majorité des simulations est faite avec 20 paquets contenant chacun 400 blocs OFDM.

## 4.2 Simulations des techniques adaptatives

Nous allons commencer par la catégorie la moins nombreuse de techniques, celles qui sont adaptatives. Nous y retrouvons le LMS et le RLS. Mentionnons qu'au cours de ce projet, nous avons tenté de simuler un réseau de neurones, mais le manque de documentation à son sujet dans les applications MIMO nous a empêchés de valider cette technique. Elle aurait été intéressante à comparer aux deux autres, et nous aurait permis de simuler l'effet de non-linéarités, mais l'information disponible n'a pas permis de la compléter.

Nous allons débiter avec des simulations mettant en évidence, pour chacune des deux techniques présentées, l'importance du choix de la quantité de données d'adaptation. Pour le LMS d'abord, nous obtenons la figure 11.

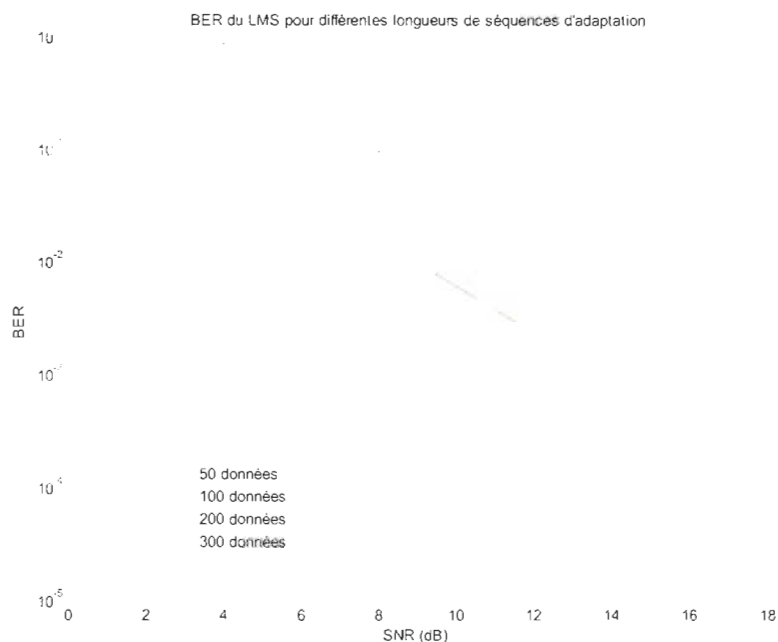


Figure 11 : BER du LMS

Cette simulation a été faite avec deux antennes de transmission et quatre de réception. Au fil de simulations, un pas d'adaptation de 0.02 a été déterminé comme étant optimal pour le LMS dans cette plate-forme; c'est ce que nous utilisons ici. Nous observons que pour des SNR faibles (8dB et moins), les performances varient très peu selon le nombre de données d'adaptation que nous avons utilisées. Toutefois, peu importe le nombre utilisé, le LMS finit par se heurter à un plancher (on le voit pour 50, ici). Le nombre de données d'adaptation influence donc assez sur les performances avec des bons SNR, car un nombre trop faible produira des erreurs, même si le bruit est presque inexistant. Pour la figure 12, faisons la même simulation, mais cette fois-ci avec le RLS.

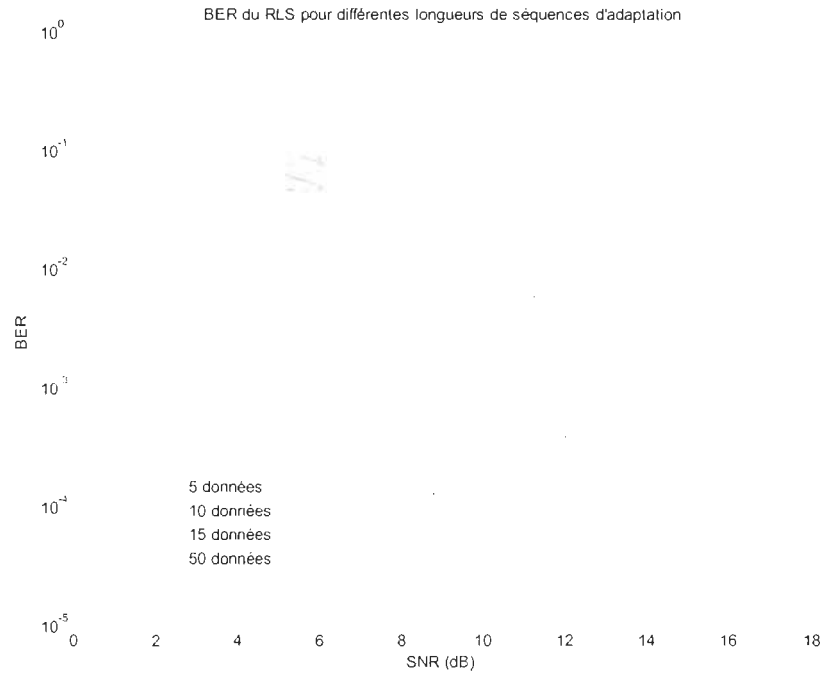


Figure 12 : BER du RLS

La différence du nombre de données d'adaptation nécessaires entre le LMS et le RLS est assez frappante. Pour 10 données d'adaptation, le RLS donne pratiquement la même courbe que le LMS avec 200. D'ailleurs, à partir de 10 données, les performances ne varient pratiquement plus. Cela signifie donc que l'adaptation est complétée très rapidement, contrairement au LMS. Précisons que nous avons utilisé un facteur d'oubli de 0.95. Pour la suite des simulations visant à comparer le LMS au RLS, nous utiliserons des nombres de données à partir desquels nous pouvons considérer l'adaptation comme pratiquement terminée, soit 10 pour le RLS et 200 pour le LMS. Observons sur la figure 13 l'impact du nombre d'antennes :

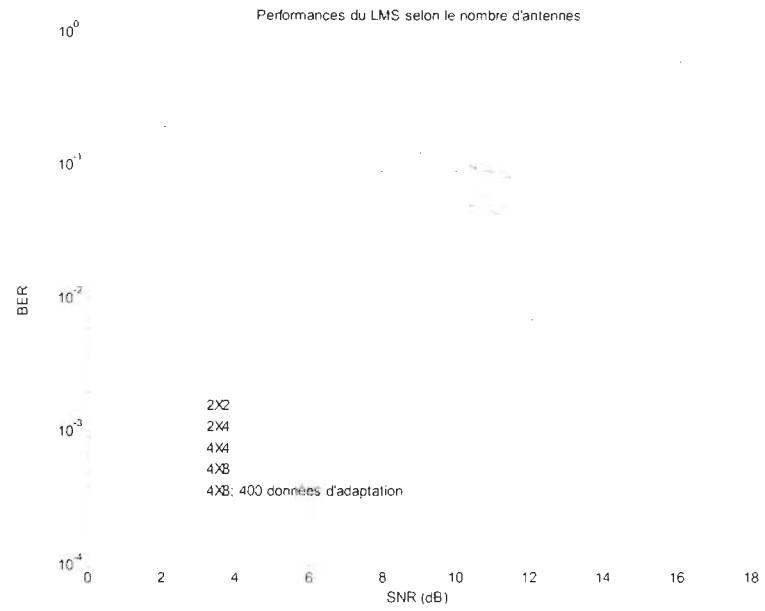


Figure 13 : Performances du LMS selon le nombre d'antennes

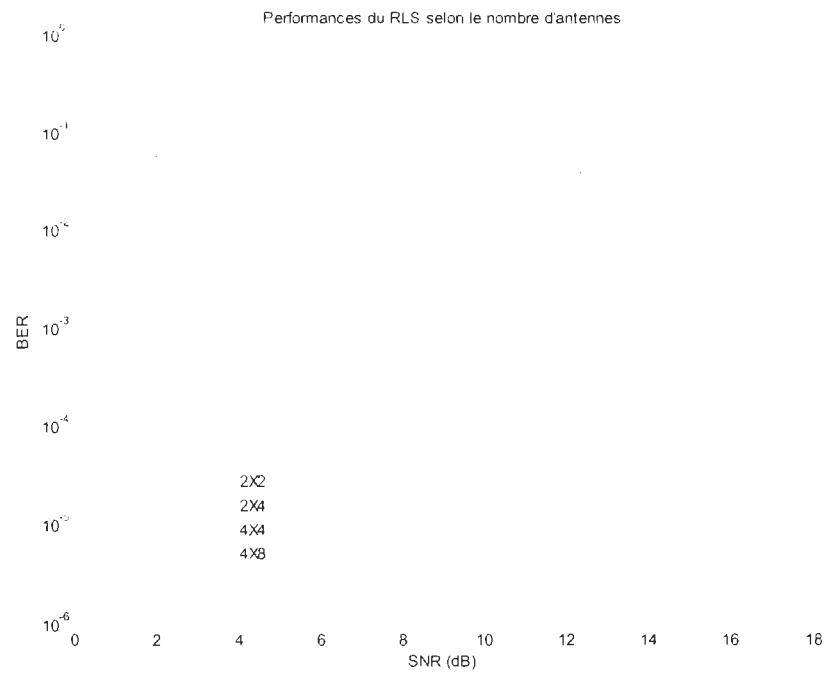


Figure 14 : Performances du RLS selon le nombre d'antennes



Dans ces figures, nous observons que le LMS offre ses meilleures performances pour 2 antennes de transmission et 4 de réception. Le  $4 \times 8$  a été simulé pour 200 et 400 données d'adaptation, pour s'assurer que la baisse de performance ne provenait pas d'une adaptation incomplète. Les deux courbes étant pratiquement superposées, cela raye cette hypothèse des explications possibles. Nous devons donc conclure que la simplicité de cet algorithme semble lui nuire lorsque les nombres d'antennes augmentent, et par conséquent, la diversité). Pour  $2 \times 2$  et  $4 \times 4$ , il n'est pas vraiment efficace. Enfin, pour  $2 \times 4$ , les résultats sont très bons, pratiquement identiques à ceux du RLS. Rappelons, comme nous avons dit, que la séquence servant à calibrer le LMS est de 200 données tandis que celle du RLS n'est que de 10. Donc oui, les courbes sont semblables, mais les performances du LMS sont loin d'atteindre celles du RLS, ce qui est prévisible.

Nous voyons aussi que le RLS gère beaucoup mieux le  $4 \times 8$ . En effet, contrairement au LMS pour lequel le  $4 \times 8$  donne un BER de loin supérieur au  $2 \times 4$ , le RLS offre d'excellentes performances dans cette configuration. Avant environ 14dB, le BER est légèrement supérieur au  $2 \times 4$ , mais après, pour les signaux sans trop de bruit, le taux d'erreur chute abruptement. Les systèmes  $2 \times 2$  et  $4 \times 4$  donnent des résultats du niveau de ce que le LMS nous a donné. C'est un fait que les systèmes avec plus d'antennes de réception que de transmission sont réputés offrir de meilleures performances. Les courbes ne font que le confirmer.

Nous pouvons tirer quelques conclusions pour les systèmes adaptatifs. Tout d'abord, le LMS peut être un choix viable par rapport au RLS, mais pas dans n'importe quelles conditions. Pour un système  $2 \times 4$ , où le canal subit peu de changements, il serait un très bon candidat potentiel. En effet, limiter les changements au niveau du canal permettrait

d'envoyer la séquence d'entraînement moins souvent, ce qui compenserait pour sa longueur. Toutefois, pour des systèmes avec plus d'antennes, le RLS est visiblement un meilleur choix. Dans les deux cas, les systèmes comptant autant d'antennes de transmission que de réception sont à déconseiller. Avant de passer, à autre chose, rappelons que ces deux techniques ont l'avantage de ne pas requérir d'estimateur de canal.

### **4.3 Simulation des techniques non adaptatives**

Passons maintenant aux techniques qui ne demandent pas de séquence d'entraînement. Nous avons simulé les différentes techniques pour plusieurs nombres d'antennes, pour comparer leurs performances. Débutons avec les plus simples, soit le ZF et le MMSE, présentés à la figure 15.

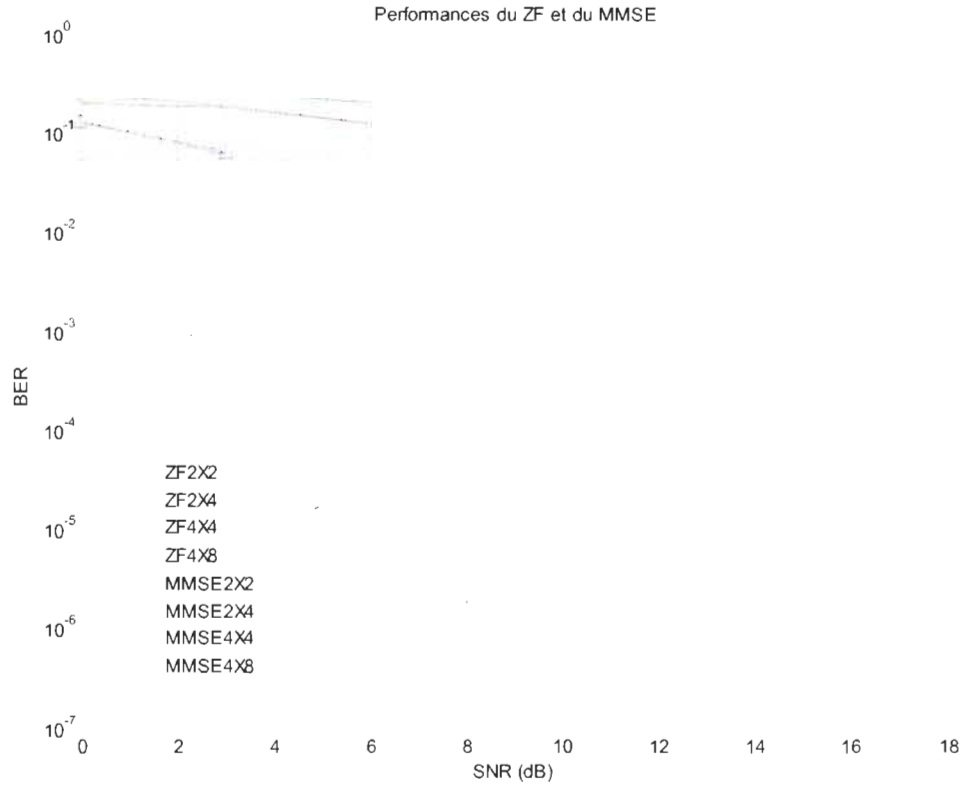


Figure 15 : Performances du ZF et du MMSE

Au premier coup d'œil, nous pouvons voir que toutes les courbes ZF et MMSE, pour un même nombre d'antennes, sont pratiquement superposées. Il n'y a rien de surprenant, et l'explication s'en trouve dans l'équation 3.4. Effectivement, la seule différence entre l'équation du ZF et celle du MMSE est l'addition d'un facteur proportionnel au rapport des puissances du signal et du bruit. À partir d'un SNR de 3dB, ce rapport devient presque négligeable, et a de moins en moins d'impact sur le résultat. C'est exactement ce que nous voyons ici : des courbes qui sont distinctes au début, mais qui se rejoignent rapidement pour ne former qu'une. Comme nous l'avons observé avec les techniques adaptatives, les performances sont bien meilleures lorsque le nombre d'antennes de transmission est

supérieur à celui des antennes de réception. Aucune des deux n'a des problèmes avec la diversité telle qu'expérimenté avec le LMS.

Répetons l'expérience avec les techniques BLAST, soit le ZF-VBLAST et le MMSE-VBLAST (Figure 16).

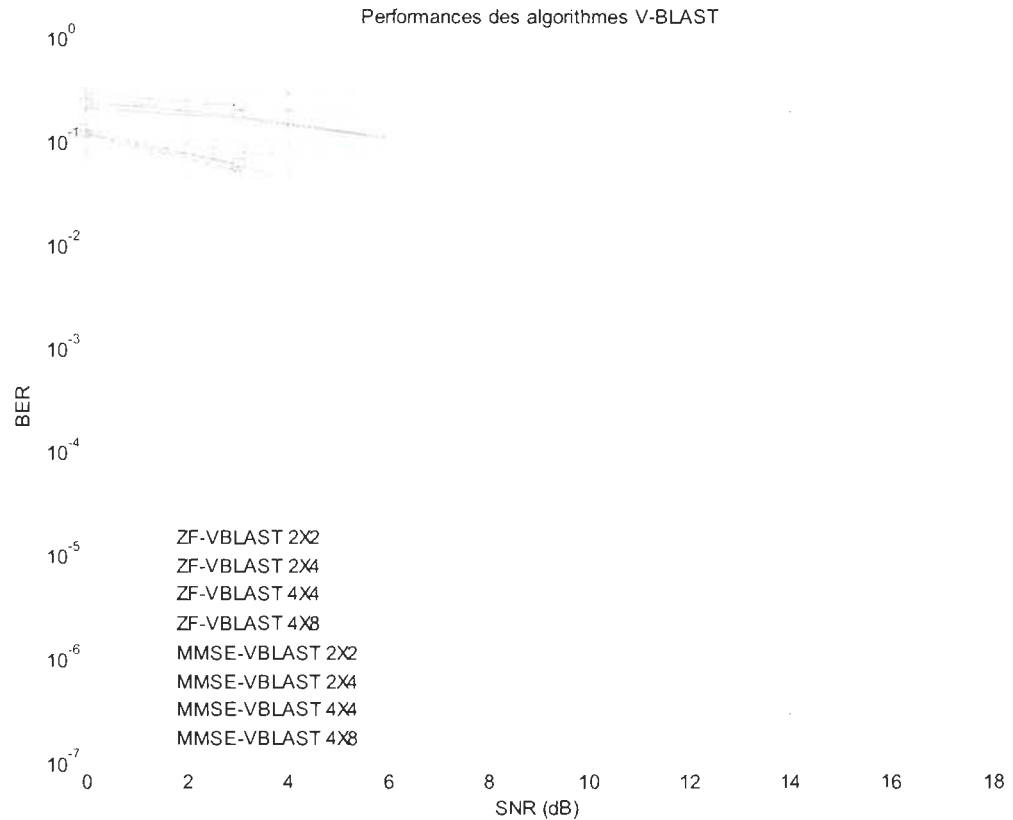


Figure 16 : Performances des algorithmes V-BLAST

Sur ce graphique, nous voyons qu'il n'y a pas de réelle surprise, et que les résultats ressemblent à ce que nous nous attendions. Nous voyons que les algorithmes BLAST offrent d'excellentes performances dans les configurations 2×4 et 4×8, atteignant des taux d'erreur très faible à des SNR relativement bas. Comme précédemment, la différence entre MMSE et ZF est assez mince et n'est présente qu'au début. Les différents algorithmes

seront comparés entre eux après avoir affiché leurs résultats individuellement, afin de faciliter la lecture et l'interprétation de ceux-ci.

Poursuivons avec les algorithmes SQRD. De ce que nous en avons dit plus tôt, nous sommes ici en droit de nous attendre d'obtenir des résultats très semblables au précédents.

Voyons sur la Figure 17 ce que donnent les simulations :

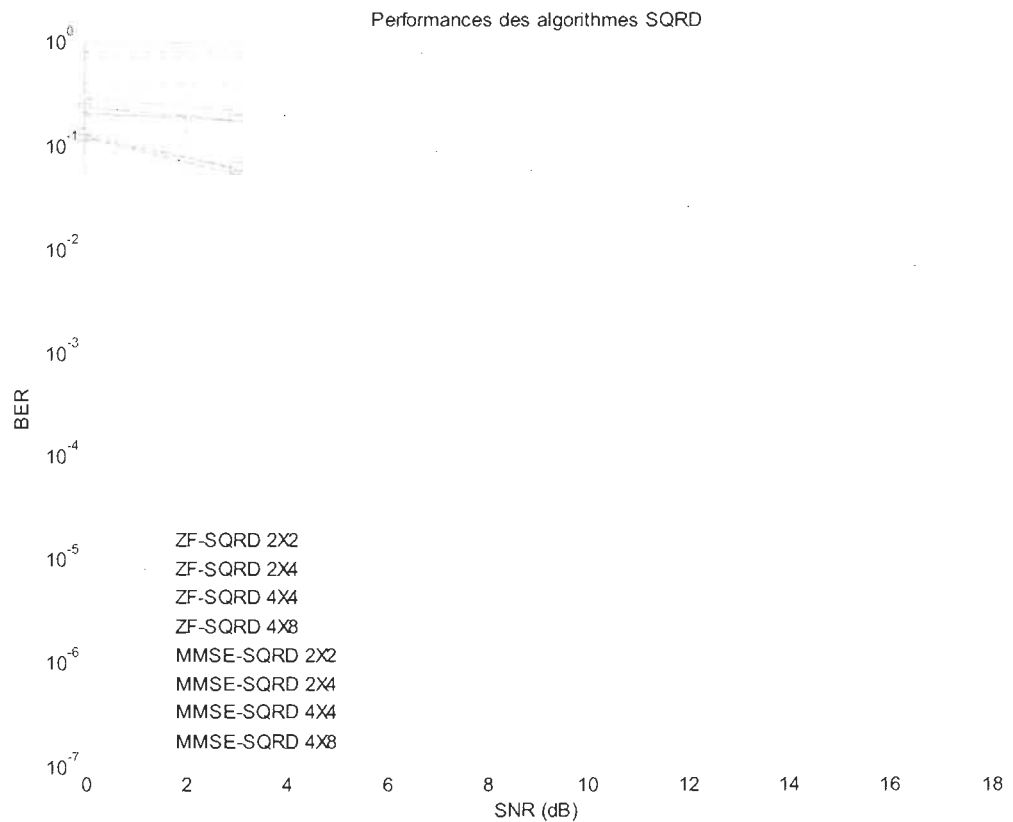


Figure 17 : Performances des algorithmes SQRD

Les courbes obtenues pour les taux d'erreurs correspondent très bien à nos attentes. En fait, au premier coup d'œil, elles semblent calquées sur celles du V-BLAST. Pourtant, nous devons rappeler qu'il s'agit d'une version allégée du V-BLAST au niveau des calculs.

Montrons sur la figure 18 une comparaison plus large de plusieurs des techniques que nous venons de voir :

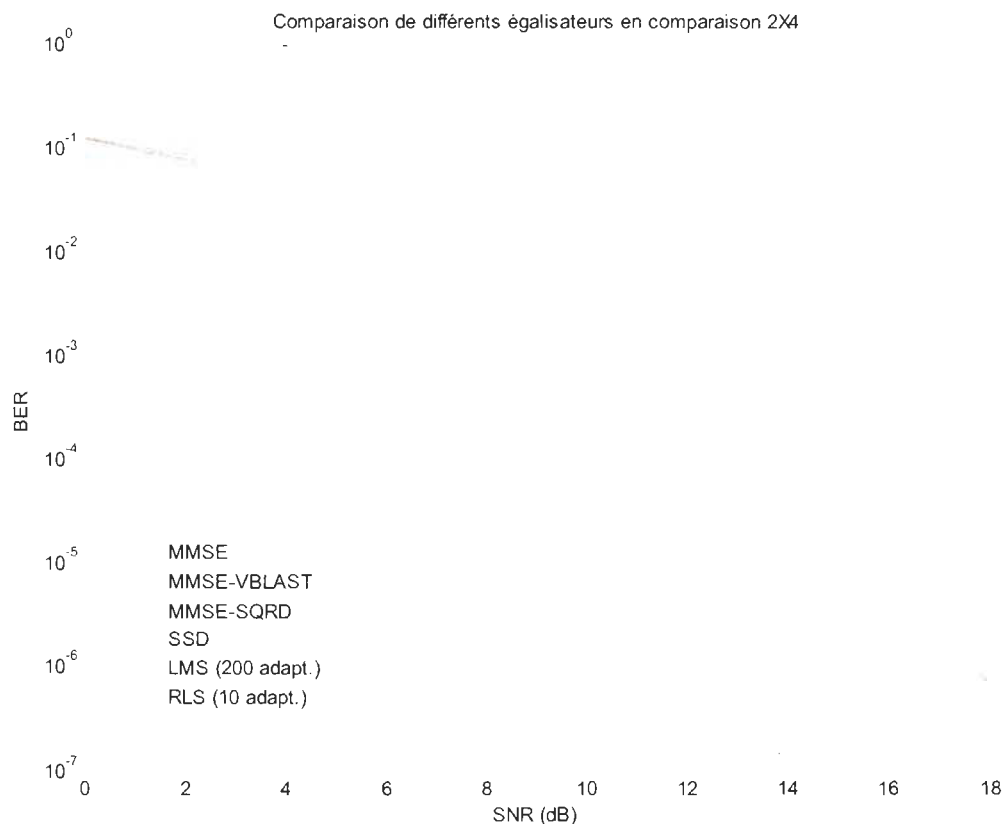


Figure 18 : Comparaison d'égaliseurs en configuration 2X4

Ce graphique nous permet de comparer toutes les techniques que nous avons vues, à l'exception des ZF, qui étaient, sauf pour de très faibles SNR, pratiquement identiques à leurs équivalents MMSE. Nous avons aussi inclus les techniques adaptatives, afin d'avoir une vue d'ensemble. La configuration de deux antennes de transmission et quatre de réception a été choisie, car toutes les techniques ont bien fonctionné dessus. Ce graphique nous montre plusieurs éléments intéressants. Tout d'abord, les techniques adaptatives, même lorsque les poids ont complètement fini de converger, n'atteignent jamais les

performances des techniques non adaptatives. Ensuite, nous observons que le SSD a pris la place que nous lui attendions, soit la meilleure performance, puisqu'il donne des résultats optimaux. Il peut donc servir de référence pour les autres. Le V-BLAST montre qu'il mérite bien sa réputation, en gardant un écart de 1dB ou moins avec le SSD. Les performances s'en rapprochent bien. De la même façon, le SQRD approche beaucoup du V-BLAST, et, par le fait même, du SSD. Rappelons que l'analyse présentée ici résulte purement de l'observation des courbes; elle est purement mathématique. Une analyse plus détaillée prenant d'autres facteurs en compte conclut ce chapitre.

#### 4.4 Complexité algorithmique

Bien que tout ce mémoire soit surtout basé sur des simulations, plutôt que sur l'implémentation, nous devons ne pas perdre de vue la réalité : une technique, aussi bonne soit-elle, ne vaudra rien en pratique si elle n'est pas utilisable dans les applications d'intérêt. Il y a plusieurs façons de compter les opérations; certaines plus approximatives, d'autres très précises. Toutefois, la variété des techniques d'égalisation que nous avons analysées fait que le calcul précis des ressources utilisées n'est pas évident. Nous allons donc les comparer sommairement avec la notation grand O, qui indique l'ordre maximal des opérations qui s'effectuent. Cette comparaison est présentée dans le Tableau 1

Ce tableau donne une idée générale de la complexité des différentes techniques d'égalisation que nous avons vues. Le ML a été inclus pour montrer pourquoi il est pratiquement impossible à implanter. Sa complexité croît exponentiellement avec le nombre d'antennes en jeu, ainsi que le nombre de symboles QAM possible. Il s'agit donc purement d'une solution mathématique optimale. Pour ce qui est des techniques adaptatives, les deux sont du même ordre, qui est d'ailleurs le plus faible. Évidemment,

leur complexité n'est pas la même pour autant, mais l'absence d'inversion de matrices joue définitivement en leur faveur. Elles sont donc moins lourdes en calcul que les techniques non adaptatives. Nous observons que les techniques SQRD ont un ordre de complexité inférieur aux techniques BLAST, qui est le gain acquis en évitant l'inversion de matrice. Enfin, notons que malgré son ordre semblable aux techniques BLAST, le SSD a une complexité beaucoup plus importante. Néanmoins, il ne se rapproche même pas du ML en termes de complexité, et réussit à donner aussi des résultats optimaux, en assumant que le rayon du cercle est bien ajusté.

Algorithme	Ordre
LMS	$O(N^2)$
RLS	$O(N^2)$
ZF/MMSE	$O(N^3)$
ZF/MMSE-VBLAST	$O(N^4)$
ZF/MMSE-SQRD	$O(N^3)$
SSD	$O(N^4)$
ML	$O(N^N)$

Tableau 1 : Complexité des algorithmes

#### 4.5 Analyse finale et recommandations d'égaliseurs

Que pouvons-nous dire sur ces techniques d'égalisation après ces analyses et réflexions? Comme nous pouvons nous en douter, il n'y a pas de technique parfaite. Pour des milieux où il y a peu de variations de conditions (bruit et canal), les techniques adaptatives peuvent s'avérer intéressantes, puisqu'elles sont les moins complexes. Nous devons considérer l'apprentissage considérablement plus lent du LMS lors d'un éventuel



choix, mais il peut être utilisé si les conditions changent vraiment peu. Cela permettrait de faire l'adaptation moins souvent, et limiterait l'impact négatif qu'est sa vitesse d'adaptation. Sinon, il y a toujours moyen de se tourner vers le RLS, qui peut se calibrer très rapidement. Il est intéressant de noter que ces techniques ne requièrent pas d'estimateur de canal, contrairement aux autres techniques traitées. Cela sauve des calculs que nous n'avons pas pris en compte lors de ce projet, puisque nous assumions toujours que le canal avait été préalablement parfaitement estimé. Les solutions adaptatives que nous avons étudiées sont donc de loin les plus simples à implémenter.

Pour les égaliseurs non adaptatifs analysés, il y en a un qui ressort avantageusement du lot : le SQRD. En effet, puisqu'il a un ordre de complexité inférieur au V-BLAST et au SSD, nous pourrions nous attendre d'obtenir des résultats plus près du ZF ou du MMSE. Pourtant, il a atteint le but fixé : approcher les performances du V-BLAST tout en réduisant considérablement la charge de calcul. Il offre donc un excellent rapport de performance. Nous remarquons que le ZF et le MMSE sont peu intéressants à utiliser en pratique, car leurs performances approchent celles des techniques adaptatives, qui sont plus simples. Une bonne partie de l'intérêt du V-BLAST est le fait qu'il soit bien connu et documenté. Cela peut être une raison pour motiver son choix. Toutefois, nous pouvons constater que la charge de calcul supplémentaire ne vaut pas vraiment la faible différence de performance qu'il a avec le SQRD. Le SSD sert de référence dans les graphiques, mais il reste relativement lourd à utiliser en pratique, certes, beaucoup moins que le ML ou le SD classique, mais plus que toute autre technique analysée ici. Enfin, observons que la différence de performance entre le ZF et le MMSE, pour les techniques, est relativement faible, sauf dans des milieux très bruités. L'ajout du MMSE ajoute peu de charge de calcul.

Toutefois, pour bien l'utiliser, il faut avoir une certaine connaissance du bruit qui sera ajouté au signal, sinon nous risquons d'obtenir de moins bonnes performances.

Rappelons enfin que les techniques non adaptatives ne seront d'aucune utilité si le canal n'est pas connu. Il faut donc s'assurer d'avoir un estimateur de canal qui nous permet d'atteindre le niveau de performance désiré, selon la technique que nous choisissons.

## Chapitre 5 - Conclusion

Au cours de ce mémoire, nous avons effectué une analyse comparative de techniques variées d'égalisation pour des systèmes MIMO-OFDM. Nous allons donc faire un bref retour sur le projet, pour terminer avec des possibilités de développements futurs.

### 5.1 Retour sur le contenu

Nous pouvons conclure que l'objectif principal établi au départ a été accompli, soit d'effectuer une étude comparative de plusieurs méthodes d'égalisation pour des systèmes de télécommunications de type MIMO-OFDM. Pour se faire, nous avons complétés les objectifs secondaires que nous nous étions fixés. Le tout est décrit dans les lignes qui suivent.

Tout d'abord, une recherche bibliographique dans les articles scientifiques nous a fourni les informations pour établir la problématique, et a fait ressortir plusieurs techniques d'égalisation, linéaires ou non, adaptatives ou non. Nous pouvons présumer que certaines de ces techniques n'avaient pas été testées avec des systèmes MIMO-OFDM, puisqu'il n'y avait aucune information disponible sur ce sujet.

Nous avons exploré individuellement les concepts d'OFDM et de MIMO, afin d'en venir à traiter du système MIMO-OFDM, dont nous avons décrit les principaux éléments. Puis, nous avons approfondi chacune des techniques que nous avons retenues pour l'étude. Les techniques moins communes (SQRD et SSD) ont été développées plus en détails, pour

compenser la difficulté de trouver l'information nécessaire. De plus, ce sont les deux techniques qui contenaient des erreurs dans leurs articles d'origine. Un travail supplémentaire a dû être effectué pour les identifier et corriger, ce qui n'était pas évident vu l'absence d'autres articles de référence.

Ensuite nous avons détaillé la plate-forme que nous avons développée. C'est une plate-forme extrêmement flexible qui nous permet de faire toutes les simulations voulues. Dans notre cas, nous voulions des conditions identiques pour chacune des techniques étudiées afin d'avoir des résultats ayant une bonne valeur scientifique pour la comparaison.

Enfin, nous avons fait des simulations des techniques de détection MIMO-OFDM sur cette plate-forme, afin de faire ressortir les principales caractéristiques des différents algorithmes. Nous ne pouvons dire qu'une technique surpasse toutes les autres, mais celle qui a attiré notre attention est le SQRD, qui offre un compromis plus que satisfaisant entre simplicité algorithmique et performances. Ce choix reste circonstanciel, car dans certaines situations avec peu d'interférences où le canal varie peu, le LMS pourrait très bien suffire, par exemple. Nous en revenons toujours au compromis de la complexité que nous pouvons nous permettre et des performances que nous désirons obtenir.

## **5.2 Recommandations**

Cette recherche ouvre la porte à plusieurs possibilités. Rappelons que tout au long de ce mémoire, nous avons considéré une connaissance parfaite du canal pour les techniques non adaptatives. Cette hypothèse est correcte dans le cas d'une étude comparative, mais en ajoutant un estimateur de canal, il serait possible de simuler des résultats beaucoup plus près de ceux de la réalité. Cela constituerait une suite logique à ce projet, et pourrait être la

première étape vers la conception d'un récepteur MIMO-OFDM complet. Évidemment, nous n'avons pas non plus simulé toutes les techniques, et certaines ont même certainement vu le jour durant ce projet. La plate-forme pourrait bien sûr être réutilisée pour les tester.

## Références

- [ALD01] Al-Dhahir, N., "FIR channel-shortening equalizers for MIMO ISI channels," IEEE Transactions on Communications, Volume 49, Issue 2, pp. 213-218, Feb 2001.
- [BAH04] Bahai, A. R. S.; Saltzberg, B.R.; Ergen, M., "Multi Carrier Digital Communications: Theory and Applications of OFDM," Springer, 2004.
- [BAR00] Baro, S.; Bauch, G.; Pavlic, A.; Semmler, A., "Improving BLAST performance using space-time block codes and turbo decoding", Proc. GLOBECOM'00, Vol. 2, pp.1067–1071, 2000.
- [BOL02] Bolcskei, H.; Gesbert, D.; Paulraj, A.J., "On the capacity of OFDM-based spatial multiplexing systems," IEEE Transactions on Communications, Volume 50, Issue 2, pp.225-234, Feb. 2002.
- [BOR04] Boronka, A.; Speidel, J., "Analysis and results on optimized MIMO detection with threshold-based adaptive cancellation," 2004. IEEE 59th Vehicular Technology Conference (VTC 2004-Spring), volume 2, pp. 814–818, 17-19 May 2004.
- [CHA02] Chan, A.M.; Inkyu Lee; "A new reduced-complexity sphere decoder for multiple antenna systems", 2002 IEEE International Conference on Communications (ICC 2002), vol. 1, pp. 460 – 464, 28 April-2 May 2002.
- [CHO00] Choi W.-J.; Negi, R.; Cioffi, J.M., "Combined ML and DFE decoding for V-BLAST system", Proc. ICC'00 Vol. 3, pp. 1243–1248, 2000.
- [COL02] Colieri, S.; Ergen, M.; Puri, A.; Bahai, A., "A study of channel estimation in OFDM systems," IEEE 56th Vehicular Technology Conference Proceedings. (VTC 2002-Fall), Volume 2, 24-28 Sept. 2002, pp. 894–898, 2002.
- [ELB05] El-Barbary, K.; Alneyadi, H.M.; "Comparison of the behavior of MMSE detection scheme for DS-CDMA and MC-CDMA," 2005 Second IFIP International Conference on Wireless and Optical Communications Networks (WOCN 2005), pp. 490–495, 6-8 March 2005.
- [FOS96] G.J. Foschini, « Layered Space-Time Architecture for Wireless Communication in a Fading Environment When Using Multiple Antennas», Bell Laboratories Technical Journal, Vol.1, #2, 1996, pp. 41–59.

- [GOL03] Goldsmith, A.; Jafar, S.A.; Jindal, N.; Vishwanath, S., "Capacity limits of MIMO channels," IEEE Journal on Selected Areas in Communications, volume 21, issue 5, pp. 684-702, June 2003.
- [HAR97] Hara, S.; Prasad, R., "Overview of multicarrier CDMA," IEEE Communications Magazine, vol. 35, No. 12, pp. 126-133, Dec. 1997.
- [HAY96] S. Haykin, "Adaptive Filter Theory," 3rd ed., Prentice-Hall, 1996.
- [HAY01] S. Haykin, "Communication Systems," 4th ed., John Wiley and Sons, 2001.
- [HON02] Honkasalo, H. and Al., "WCDMA and WLAN for 3G and beyond," IEEE Wireless Communications, volume 9, issue 2, pp.14-18, April 2002.
- [KAN04] Kanemaru, H.; Ohtsuki, T., "Interference cancellation with diagonalized maximum likelihood decoder for space-time/space-frequency block coded OFDM," 2004 IEEE 59th Vehicular Technology Conference (VTC 2004-Spring), volume 1, pp.525-529, 17-19 May 2004.
- [KUR04] Kuropatwinski, W.; Lillie, F.; Geng, N.; Jondral, F.K., "Low-cost MIMO real-time demonstrator: concept and first results," 2004 ITG Workshop on Smart Antennas, pp.25-29, 2004.
- [LAN04] Lang, S.; Rao R.; Daneshrad, B., "Design and development of a 5.25 GHz software defined wireless OFDM communication platform," IEEE Commun. Mag., vol. 42, no. 6, pp. 6-12, June 2004.
- [LEU03] Leus, G; Barhumi I.; Moonen M., "Per-tone equalization for MIMO-OFDM systems," IEEE International Conference on Communications 2003 (ICC '03) volume 4, pp. 2345-2349, 11-15 May 2003.
- [LI04] Li, D.; Cai, L.; Yang, H., "New iterative detection algorithm for V-BLAST," 2004 IEEE 60th Vehicular Technology Conference (VTC2004-Fall), Volume 4, 26-29, pp. 2444-2448, Sept. 2004.
- [LI05] Li, X.; Cao. X., "Low complexity signal detection algorithm for MIMO OFDM systems," Electronics Letters, volume 41, issue 2, pp. 83-85, 20 Jan. 2005.
- [LIU03] Liu, C.H., "On the design of OFDM signal detection algorithms for hardware implementation," 2003 IEEE Global Telecommunications Conference (GLOBECOM '03), Volume 2, 1-5, pp. 596-599, Dec. 2003.
- [MED06] Medles, A.; Slock, D. T. M., "Achieving the Optimal Diversity-Versus Multiplexing Tradeoff for MIMO Flat Channels With QAM Space-Time Spreading and DFE Equalization," IEEE Transactions on Information Theory, vol. 52, issue 12, pp. 5312-5323, Dec. 2006.

- [OUA02] Ouameur, M.A., "Modélisation et estimation de canaux pour les communications sans fil," Mémoire de maîtrise, Université du Québec à Trois-Rivières, Avril 2002.
- [RAO04] Rao, R. M. and Al., "Multi-antenna testbeds for research and education in wireless communications," IEEE Communication Magazine, vol. 42, no. 12, pp. 41–50, December 2004.
- [RAP96] T. S. Rappaport, "Wireless Communications: Principles and Practice," Prentice-Hall, 1996.
- [SAY03] Sayed, A.H., "Fundamentals of Adaptive Filtering", Wiley-IEEE Press, 2003.
- [SHE03] Shen, C.; Zhang, H.; Dai L.; Zhou, S. "Detection algorithm improving V-BLAST performance over error propagation", Electronics Letters, Vol. 39 #13, juin 2003.
- [SMI02] Smith, C.; Collins, D., "3G Wireless Network," The McGraw-Hill Companies Inc., 2002.
- [SON05] Song, B.; Zhang W.; Gui L., "Iterative Joint Channel Estimation and Signal Detection in MIMO OFDM Systems," 2005 International Conference on Wireless Communications, Networking and Mobile Computing, Proceedings volume 1, pp. 39–43, 23–26 Sept. 2005.
- [SUN01] Sun, Y. "Bandwidth-efficient wireless OFDM," IEEE Journal on Selected Areas in Communications, volume 19, issue 11, pp. 2267–2278, Nov. 2001.
- [SUR06] Suryanegara, M.; Hutabarat E.R.; Gunawan, D., "The Interference on WCDMA System in 3G Coexistence Network," 2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications, pp.1–5, Sept. 2006.
- [TAC02] Tachikawa, K., "W-CDMA Mobile Communications System", John Wiley and Sons, 2002.
- [TAN05] Tang, T.; Heath. Jr, R.W., "Space-time interference cancellation in MIMO-OFDM systems," IEEE Transactions on Vehicular Technology, volume 54, issue 5, pp. 1802–1816, sept. 2005.
- [UTH04] Uthansakul, P.; Bialkowski, M.E., "Multipath signal effect on the capacity of MIMO, MIMO-OFDM and spread MIMO-OFDM," 2004 15th International Conference on Microwaves, Radar and Wireless Communications (MIKON-2004), volume 3, pp. 989–992, 17–19 May 2004.
- [WAN04] Wang, J.; Daneshrad, B., "Performance of linear interpolation-based MIMO detection for MIMO-OFDM systems," 2004 IEEE Wireless Communications and Networking Conference (WCNC 2004), vol. 2, pp. 981–986, March 2004.



- [WAN05] Wang, J.; Daneshrad, B., "A comparative study of MIMO detection algorithms for wideband spatial multiplexing systems," 2005 IEEE Wireless Communications and Networking Conference, vol. 1, pp. 408–413, 13-17 March 2005.
- [WIN06] Windpassinger, C.; Lampe, L.; Fischer, R.F.H.; Hehn, T., "A performance study of MIMO detectors," IEEE Transactions on Wireless Communications, volume 5, issue 8, pp.2004-2008, Aug. 2006 .
- [WOL98] Wolniansky, P.W.; Foschini, G.J.; Golden, G.D.; Valenzuela, R.A., "V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel," 1998 URSI International Symposium on Signals, Systems, and Electronics (ISSSE 98), pp. 295-300, 29 Sept.-2 Oct. 1998.
- [WUB01] Wubben, D.; Bohnke, R.; Rinas, J.; Kuhn, V.; Kammeyer, K.D., "Efficient algorithm for decoding layered space-time codes," Electronics Letters, vol. 37, Issue 22, pp.1348-1350, 25 Oct 2001.
- [WUB03] Wubben, D.; Bohnke, R.; Kuhn, V.; Kammeyer, K.-D., "MMSE extension of V-BLAST based on sorted QR decomposition," IEEE 58<sup>th</sup> Vehicular Technology Conference (VTC 2003-Fall), Vol. 1, pp.508 – 512, 6-9 Oct. 2003.
- [YAC02] Yacoub, M. D., "Wireless technology : protocols, standards, and techniques," CRC Press, 2002.
- [YAN05] Yang, H., "A Road to Future Broadband Wireless Access: MIMO-OFDM-Based Air Interface," IEEE Communications Magazine, pp.53-60, Jan. 2005.
- [YAN06] Yang, L.L.; Hua, W.; Hanzo, L., "Multiuser detection assisted time- and frequency-domain spread multicarrier code-division multiple-access," IEEE Transactions on Vehicular Technology, volume 55, issue 1, pp. 397-404, Jan. 2006.
- [ZAN05] Zanella, A.; Chiani, M.; Win, M.Z., "MMSE reception and successive interference cancellation for MIMO systems with high spectral efficiency," IEEE Transactions on Wireless Communications, volume 4, issue 3, pp.1244–1253, May 2005.
- [ZHE03] Zheng, L.; Tse, D.N.C., "Diversity and multiplexing: a fundamental tradeoff in multiple antenna channels," IEEE Transactions on Information Theory, vol. 49, pp. 1073–1096, May 2003.